

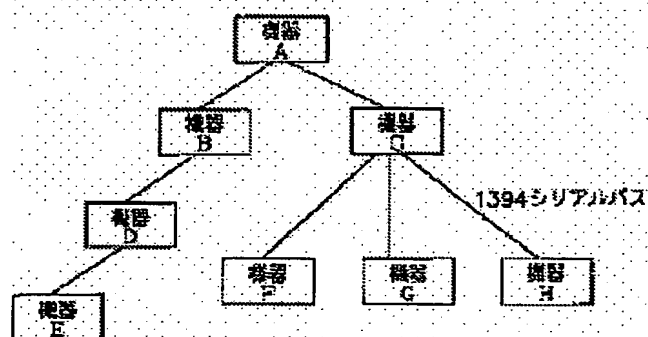
DATA PROCESSING METHOD, DATA PROCESSOR, PRINTER AND STORAGE MEDIUM

Patent number: JP10283136
 Publication date: 1998-10-23
 Inventor: KATANO KIYOSHI; NAKAMURA ATSUSHI
 Applicant: CANON KK
 Classification:
 - international: G06F3/12; B41J29/38; G06F13/00
 - european:
 Application number: JP19970089826 19970408
 Priority number(s): JP19970089826 19970408

Report a data error here

Abstract of JP10283136

PROBLEM TO BE SOLVED: To provide a data processing method with high extensibility at the time of sending out data through a serial bus by giving a specified protocol priority at the time of selecting a protocol to be executed from the plural kinds of the protocols following an initial protocol. **SOLUTION:** Equipments A-H are provided and A and B, A and C, B and D, D and E, C and F, C and G and C and H are respectively connected by the twist pair cable of a 1394 serial bus. The respective equipments A-H are respectively provided with intrinsic IDs. In such a system, the initial protocol unrelated to the kind of the protocol is executed, and at the time of selecting the protocol to be executed from the plural kinds of the protocols following the initial protocol, the specified protocol is given priority. In this case, the specified protocol is the protocol of the largest number of supporting devices in the devices connected onto a common serial bus.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-283136

(43) 公開日 平成10年(1998)10月23日

(51) Int.Cl. ⁶	識別記号	F I	
G 0 6 F 3/12		G 0 6 F 3/12	A
			D
B 4 1 J 29/38		B 4 1 J 29/38	Z
// G 0 6 F 13/00	3 5 7	G 0 6 F 13/00	3 5 7 A

審査請求 未請求 請求項の数31 O L (全 19 頁)

(21) 出願番号 特願平9-89826

(22) 出願日 平成9年(1997)4月8日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 片野 清

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(72) 発明者 中村 敦

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

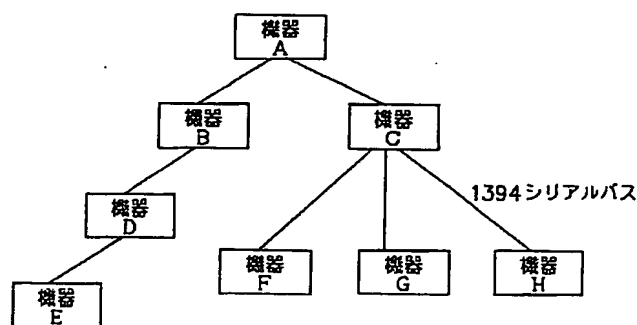
(74) 代理人 弁理士 國分 孝悦

(54) 【発明の名称】 データ処理方法、データ処理装置、プリンタ及び記憶媒体

(57) 【要約】

【課題】 シリアルバスを介してデータを送出する際に、拡張性の高いデータ処理方法、データ処理装置、プリンタ及び記憶媒体を提供する。

【解決手段】 複数のプロトコルから1つのプロトコルを選択して使用する場合に、プロトコルの変更の頻度を小さくする。これにより、プロトコルの再設定による効率の低下を防ぐことができる。また、複数デバイスのログインを許容するプロトコルを優先する。これにより、デバイスの占有を防ぐことができる。



(2)

【特許請求の範囲】

【請求項1】 共通シリアスバスを介して複数種類のプロトコルを切り換えるためのデータ処理方法であって、前記プロトコルの種類に係わらない初期プロトコルを実行し、

前記初期プロトコルに続いて前記複数種類のプロトコルから実行すべきプロトコルを選択する際には、特定のプロトコルを優先することを特徴とするデータ処理方法。

【請求項2】 前記特定のプロトコルは、前記共通シリアルバス上に接続されたデバイスのうち、サポートするデバイスの最も多いプロトコルであることを特徴とする請求項1記載のデータ処理方法。

【請求項3】 前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスは、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記特定のプロトコルとして各デバイスに通知することを特徴とする請求項2記載のデータ処理方法。

【請求項4】 前記特定のプロトコルは、複数デバイスのログインを許容するプロトコルであることを特徴とする請求項1記載のデータ処理方法。

【請求項5】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項1記載のデータ処理方法。

【請求項6】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項1記載のデータ処理方法。

【請求項7】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項1記載のデータ処理方法。

【請求項8】 共通シリアスバスを介して複数種類のプロトコルを切り換えるためのデータ処理装置であって、前記プロトコルの種類に係わらない初期プロトコルを実行する第1の実行手段と、前記初期プロトコルに続いて前記複数種類のプロトコルから実行すべきプロトコルを選択する際には、特定のプロトコルを優先する第2の実行手段とを備えることを特徴とするデータ処理装置。

【請求項9】 前記第2の実行手段は、前記共通シリアルバス上に接続されたデバイスのうち、サポートするデバイスの最も多いプロトコルを、前記特定のプロトコルとすることを特徴とする請求項8記載のデータ処理装置。

【請求項10】 前記第2の実行手段は、前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスにより、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記特定

2

のプロトコルとして各デバイスに通知することを特徴とする請求項9記載のデータ処理装置。

【請求項11】 前記第2の実行手段は、複数デバイスのログインを許容するプロトコルを前記特定のプロトコルとすることを特徴とする請求項8記載のデータ処理装置。

【請求項12】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項8記載のデータ処理装置。

10 【請求項13】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項8記載のデータ処理装置。

【請求項14】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項8記載のデータ処理装置。

【請求項15】 共通シリアスバスを介して複数種類のプロトコルを切り換えるプリンタであって、前記プロトコルの種類に係わらない初期プロトコルを実行する第1の実行手段と、

20 前記初期プロトコルに続いて前記複数種類のプロトコルから実行すべきプロトコルを選択する際には、特定のプロトコルを優先する第2の実行手段とを備えることを特徴とするプリンタ。

【請求項16】 前記第2の実行手段は、前記共通シリアルバス上に接続されたデバイスのうち、サポートするデバイスの最も多いプロトコルを、前記特定のプロトコルとすることを特徴とする請求項15記載のプリンタ。

30 【請求項17】 前記第2の実行手段は、前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスにより、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記特定のプロトコルとして各デバイスに通知することを特徴とする請求項16記載のプリンタ。

【請求項18】 前記第2の実行手段は、複数デバイスのログインを許容するプロトコルを前記特定のプロトコルとすることを特徴とする請求項15記載のプリンタ。

40 【請求項19】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項15記載のプリンタ。

【請求項20】 前記プロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする請求項15記載のプリンタ。

【請求項21】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項15記載のプリンタ。

50 【請求項22】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項15記載のプリンタ。

(3)

3

【請求項23】 前記任意のプロトコル実行後に、プリントすべき画像データを伝送する伝送手段を備えることを特徴とする請求項15記載のプリンタ。

【請求項24】 光電変換により前記画像データを得る撮影手段を備えることを特徴とする請求項23記載のプリンタ。

【請求項25】 共通シリアルバスを介して複数種類のプロトコルを切り換えるためのデータ処理を実行するためのステップが格納された記憶媒体であって、前記プロトコルの種類に係わらない初期プロトコルを実行する第1の実行ステップと、前記初期プロトコルに続いて前記複数種類のプロトコルから実行すべきプロトコルを選択する際には、特定の10 プロトコルを優先する第2の実行ステップとをコンピュータが読出可能に格納したことを特徴とする記憶媒体。

【請求項26】 前記第2の実行ステップは、前記共通シリアルバス上に接続されたデバイスのうち、サポートするデバイスの最も多いプロトコルを、前記特定の15 プロトコルとするステップを含むことを特徴とする請求項25記載の記憶媒体。

【請求項27】 前記第2の実行ステップは、前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスにより、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記20 特定のプロトコルとして各デバイスに通知するステップを含むことを特徴とする請求項26記載の記憶媒体。

【請求項28】 前記第2の実行ステップは、複数デバイスのログインを許容するプロトコルを前記特定の25 プロトコルとするステップを含むことを特徴とする請求項25記載のプリンタ。

【請求項29】 前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする請求項25記載の記憶媒体。

【請求項30】 前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする請求項25記載の記憶媒体。

【請求項31】 前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする請求項25記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、共通シリアルバスを介して、複数種類のプロトコルを切り換えるためのデータ処理方法、データ処理装置、プリンタ及び記憶媒体に関するものである。

【0002】

【従来の技術】従来より、例えば、シリアルバスを介してプリンタにデータを送出するシステムとして、様々な種類のシステムが知られている。

4

【0003】例えば、SCSI (Small Computer System Interface)、セントロニクス等、一般に広く用いられるようになったデファクトスタンダードのインターフェースを用いて、コンピュータからプリンタにデータを出力する技術が知られている。

【0004】

【発明が解決しようとする課題】しかしながら、シリアルバスを用いてデータを送出する従来のプロトコルは、メーカ固有の1つに限られていた。したがって、拡張性に欠ける、という問題が生じていた。特に、様々な種類の機器を接続するインターフェース、例えば、IEEE1394のようなインターフェースを用いて、データを出力する際には、かかる拡張性に欠けるという問題点は解決すべき大きな課題であった。

【0005】そこで、本発明は、上記の欠点を除去するために成されたもので、シリアルバスを介してデータを送出する際に、拡張性の高いデータ処理方法、データ処理装置、プリンタ及び記憶媒体を提供することを目的とする。

20 【0006】

【課題を解決するための手段】本発明は、共通シリアルバスを介して複数種類のプロトコルを切り換えるためのデータ処理方法であって、前記プロトコルの種類に係わらない初期プロトコルを実行し、前記初期プロトコルに続いて前記複数種類のプロトコルから実行すべきプロトコルを選択する際には、特定の25 プロトコルを優先することを特徴とする。また、前記特定のプロトコルは、前記共通シリアルバス上に接続されたデバイスのうち、サポートするデバイスの最も多いプロトコルであることを特徴とする。また、前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスは、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記特定の30 プロトコルとして各デバイスに通知することを特徴とする。また、前記特定のプロトコルは、複数デバイスのログインを許容するプロトコルであることを特徴とする。また、前記共通シリアルバスは、IEEE1394規格に適合したバスであることを特徴とする。また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする。また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。本発明は、共通シリアルバスを介して複数種類のプロトコルを切り換えるためのデータ処理装置であって、前記プロトコルの種類に係わらない初期プロトコルを実行する第1の実行手段と、前記初期プロトコルに続いて前記複数種類のプロトコルから実行すべきプロトコルを選択する際には、特定の35 プロトコルを優先する第2の実行手段とを備えることを特徴とする。また、前記第2の実行手段は、前記共通シリアルバス上に接続された

50

(4)

5

デバイスのうち、サポートするデバイスの最も多いプロトコルを、前記特定のプロトコルとすることを特徴とする。また、前記第2の実行手段は、前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスにより、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記特定のプロトコルとして各デバイスに通知することを特徴とする。また、前記第2の実行手段は、複数デバイスのログインを許容するプロトコルを前記特定のプロトコルとすることを特徴とする。また、前記共通シリアルバスは、IEEE 1394規格に適合したバスであることを特徴とする。また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする。また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。本発明は、共通シリアルバスを介して複数種類のプロトコルを切り換えるプリンタであって、前記プロトコルの種類に係わらない初期プロトコルを実行する第1の実行手段と、前記初期プロトコルに続いて前記複数種類のプロトコルから実行すべきプロトコルを選択する際には、特定のプロトコルを優先する第2の実行手段とを備えることを特徴とする。また、前記第2の実行手段は、前記共通シリアルバス上に接続されたデバイスのうち、サポートするデバイスの最も多いプロトコルを、前記特定のプロトコルとすることを特徴とする。また、前記第2の実行手段は、前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスにより、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記特定のプロトコルとして各デバイスに通知することを特徴とする。また、前記第2の実行手段は、複数デバイスのログインを許容するプロトコルを前記特定のプロトコルとすることを特徴とする。また、前記共通シリアルバスは、IEEE 1394規格に適合したバスであることを特徴とする。また、前記プロトコルは、インクジェットプリンタ用のプロトコルであることを特徴とする。また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする。また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。また、前記任意のプロトコル実行後に、プリントすべき画像データを伝送する伝送手段を備えることを特徴とする。また、光電変換により前記画像データを得る撮影手段を備えることを特徴とする。本発明は、共通シリアルバスを介して複数種類のプロトコルを切り換えるためのデータ処理を実行するためのステップが格納された記憶媒体であって、前記プロトコルの種類に係わらない初期プロトコルを実行する第1の実行ステップと、前記初期プロトコルに続いて前記複数種類のプロト

6

コルから実行すべきプロトコルを選択する際には、特定のプロトコルを優先する第2の実行ステップとをコンピュータが読出可能に格納したことを特徴とする。また、前記第2の実行ステップは、前記共通シリアルバス上に接続されたデバイスのうち、サポートするデバイスの最も多いプロトコルを、前記特定のプロトコルとするステップを含むことを特徴とする。また、前記第2の実行ステップは、前記共通シリアルバス上に接続されたデバイスのうちの任意のデバイスにより、プロトコル毎に、前記共通シリアルバス上のそのプロトコルをサポートするデバイスの数を検出し、サポートするデバイスの最も多いプロトコルを前記特定のプロトコルとして各デバイスに通知するステップを含むことを特徴とする。また、前記第2の実行ステップは、複数デバイスのログインを許容するプロトコルを前記特定のプロトコルとするステップを含むことを特徴とする。また、前記共通シリアルバスは、IEEE 1394規格に適合したバスであることを特徴とする。また、前記共通シリアルバスは、USB規格に適合したバスであることを特徴とする。また、前記初期プロトコルは、OSIモデルのデータリンク層よりも上位の層で実行されるプロトコルであることを特徴とする。

【0007】

【発明の実施の形態】以下、本発明の実施の形態について図面を用いて説明する。

【0008】ここで、まず、以下に説明する第1及び第2の実施の形態では、各機器間を接続するデジタルI/Fとして、IEEE 1394シリアルバスを用いているため、IEEE 1394シリアルバスについて、予め説明する。

【0009】民生用デジタルVCRやDVDプレーヤの登場に伴って、ビデオデータやオーディオデータなどのリアルタイムで、かつ高情報量のデータ転送のサポートが必要になっている。こういったビデオデータやオーディオデータをリアルタイムで転送し、パソコン（PC）に取り込んだり、またはその他のデジタル機器に転送を行なうには、必要な転送機能を備えた高速データ転送可能なインタフェースが必要になってくるものであり、そういった観点から開発されたインタフェースが、IEEE 1394-1995（High Performance Serial Bus）（以下、単に1394シリアルバスとも言う）である。

【0010】図1に1394シリアルバスを用いて構成されるネットワーク・システムの例を示す。

【0011】このシステムは、機器A、B、C、D、E、F、G、Hを備えており、A-B間、A-C間、B-D間、D-E間、C-F間、C-G間、及びC-H間をそれぞれ1394シリアルバスのツイスト・ペア・ケーブルで接続されている。この機器A～Hは、例えばパソコン、デジタルVTR、DVD、デジタルカメラ、ハ

(5)

7

ードディスク、モニタ、チューナー、モニター等である。

【0012】各機器間の接続方式は、ディジーチェーン方式とノード分岐方式とを混在可能としたものであり、自由度の高い接続が可能である。

【0013】また、各機器は各自固有のIDを有し、それぞれが認識し合うことによって1394シリアルバスで接続された範囲において、1つのネットワークを構成している。各デジタル機器間をそれぞれ1本の1394シリアルバスケーブルで順次接続するだけで、それぞれの機器が中継の役割を行い、全体として1つのネットワークを構成するものである。また、1394シリアルバス、Plug & Play機能でケーブルを機器に接続した時点で自動で機器の認識や接続状況などを認識する機能を有している。

【0014】また、図1に示したようなシステムにおいて、ネットワークからある機器が削除されたり、または新たに追加されたときなど、自動的にバスリセットを行い、それまでのネットワーク構成をリセットしてから、新たなネットワークの再構築を行なう。この機能によって、その時々ネットワークの構成を常時設定、認識することができる。

【0015】またデータ転送速度は、100/200/400Mbpsと備えており、上位の転送速度を持つ機器が下位の転送速度をサポートし、互換をとるようになっている。

【0016】データ転送モードとしては、コントロール信号などの非同期データ(Asynchronousデータ：以下、Asyncデータと言う)を転送するAsynchronous転送モードとリアルタイムなビデオデータやオーディオデータ等の同期データ(Isochronousデータ：以下、Isoデータ言う)を転送するIsochronous転送モードがある。このAsyncデータとIsoデータは各サイクル(通常1サイクル125μs)の中において、サイクル開始を示すサイクル・スタート・パケット(CSP)を転送した後、Isoデータの転送をAsyncデータより優先しつつサイクル内で混在して転送される。

【0017】つぎに、図2に1394シリアルバスの構成要素を示す。

【0018】1394シリアルバスは全体としてレイヤ(階層)構造で構成されている。図2に示したように、1394シリアルバスのケーブルとコネクタが接続されるコネクタポートがあり、その上にハードウェアとしてフィジカル・レイヤとリンク・レイヤを位置づけている。

【0019】ハードウェア部は実質的なインターフェイスチップの部分であり、そのうちフィジカル・レイヤは符号化やコネクタ関連の制御等を行い、リンク・レイヤはパケット転送やサイクルタイムの制御等を行なう。

8

【0020】ファームウェア部のトランザクション・レイヤは、転送(トランザクション)すべきデータの管理を行ない、Read、Write、Lockの命令を出す。マネージメント・レイヤは、接続されている各機器の接続状況やIDの管理を行ない、ネットワークの構成を管理する部分である。

【0021】このハードウェアとファームウェアまでが実質上の1394シリアルバスの構成である。

【0022】またソフトウェア部のアプリケーション・レイヤは使うソフトによって異なり、インタフェース上にどのようにデータをのせるか規定する部分であり、プリンタやAVCプロトコルなどが規定されている。

【0023】以上が1394シリアルバスの構成である。

【0024】つぎに、図3に1394シリアルバスにおけるアドレス空間の図を示す。

【0025】1394シリアルバスに接続された各機器(ノード)には必ず各ノード固有の、64ビットアドレスを持たせておく。そしてこのアドレスをROMに格納しておくことで、自分や相手のノードアドレスを常時認識でき、相手を指定した通信も行なえる。

【0026】1394シリアルバスのアドレッシングは、IEEE1212規格に準じた方式であり、アドレス設定は、最初の10bitがバスの番号の指定用に、次の6bitがノードID番号の指定用に使われる。残りの48bitが機器に与えられたアドレス幅になり、それぞれ固有のアドレス空間として使用できる。最後の28bitは固有データの領域として、各機器の識別や使用条件の指定の情報などを格納する。

【0027】以上が1394シリアルバスの技術の概要である。

【0028】つぎに、1394シリアルバスの特徴といえる技術の部分、より詳細に説明する。

【0029】1394シリアルバスの電氣的仕様について説明する。

【0030】図4に1394シリアルバス・ケーブルの断面図を示す。

【0031】1394シリアルバスでは接続ケーブル内に6ピン、即ち2組のツイストペア信号線の他に、電源ラインを設けている。これによって、電源を持たない機器や、故障により電圧低下した機器等にも電力の供給が可能になっている。

【0032】電源線内を流れる電源の電圧は8~40V、電流は最大電流DC1.5Aと規定されている。

【0033】なお、DVケーブルと呼ばれる規格では電源を省いた4ピンで構成されている。

【0034】DS-Link符号化について説明する。

【0035】1394シリアルバスで採用されている、データ転送フォーマットのDS-Link符号化方式を説明するための図を図5に示す。

(6)

9

【0036】1394シリアルバスでは、DS-Link (Data/Strobe Link) 符号化方式が採用されている。このDS-Link符号化方式は、高速なシリアルデータ通信に適しており、その構成は、2本の信号線が必要とする。より対線のうち1本に主となるデータを送り、他方のより対線にはストロブ信号を送る構成になっている。受信側では、この通信されるデータと、ストロブとの排他的論理和をとることによってクロックを再現する。

【0037】このDS-Link符号化方式を用いるメリットとして、8/10B変換に比べて転送効率が高いこと、PLL回路が不要となるのでコントローラLSIの回路規模を小さくできること、更には、転送すべきデータが無いときにアイドル状態であることを示す情報を送る必要が無いので、各機器のトランシーバ回路をスリープ状態にすることができることによって、消費電力の低減が図れる、などが挙げられる。

【0038】バスリセットのシーケンスについて説明する。

【0039】1394シリアルバスでは、接続されている各機器（ノード）にはノードIDが与えられ、ネットワーク構成として認識されている。

【0040】このネットワーク構成に変化があったとき、例えばノードの挿抜や電源のON/OFFなどによるノード数の増減などによって変化が生じて、新たなネットワーク構成を認識する必要があるとき、変化を検知した各ノードはバス上にバスリセット信号を送信して、新たなネットワーク構成を認識するモードに入る。

【0041】このときの変化の検知方法は、1394ポート基盤上でのバイアス電圧の変化を検知することによって行われる。

【0042】あるノードからバスリセット信号が伝達されて、各ノードのフィジカルレイヤはこのバスリセット信号を受けると同時にリンクレイヤにバスリセットの発生を伝達し、かつ他のノードにバスリセット信号を伝達する。最終的にすべてのノードがバスリセット信号を検知した後、バスリセットが起動となる。

【0043】バスリセットは、先に述べたようなケーブル抜挿や、ネットワーク異常等によるハード検出による起動と、プロトコルからのホスト制御などによってフィジカルレイヤに直接命令を出すことによっても起動する。また、バスリセットが起動するとデータ転送は一時中断され、この間のデータ転送は待たされ、終了後、新しいネットワーク構成のもとで再開される。

【0044】以上がバスリセットのシーケンスである。

【0045】ノードID決定のシーケンスについて説明する。

【0046】バスリセットの後、各ノードは新しいネットワーク構成を構築するために、各ノードにIDを与える動作に入る。このときの、バスリセットからノードID

10

D決定までの一般的なシーケンスを図6、7、8のフローチャートを用いて説明する。

【0047】図6のフローチャートは、バスリセットの発生からノードIDが決定し、データ転送が行えるようになるまでの、一連のバスの作業を示してある。

【0048】まず、ステップS101として、ネットワーク内にバスリセットが発生することを常時監視していて、ここでノードの電源ON/OFFなどでバスリセットが発生するとステップS102に移る。

【0049】ステップS102では、ネットワークがリセットされた状態から、新たなネットワークの接続状況を知るために、直接接続されている各ノード間において親子関係の宣言がなされる。ステップS103として、すべてのノード間で親子関係が決定すると、ステップS104として一つのルートが決定する。すべてのノード間で親子関係が決定するまで、ステップS102の親子関係の宣言をおこない、またルートも決定されない。

【0050】ステップS104でルートが決定されると、次はステップS105として、各ノードにIDを与えるノードIDの設定作業が行われる。所定のノード順序で、ノードIDの設定が行われ、すべてのノードにIDが与えられるまで繰り返し設定作業が行われ、最終的にステップS106としてすべてのノードにIDを設定し終えたら、新しいネットワーク構成がすべてのノードにおいて認識されたので、ステップS107としてノード間のデータ転送が行える状態となり、データ転送が開始される。

【0051】このステップS107の状態になると、再びバスリセットが発生するのを監視するモードに入り、バスリセットが発生したらステップS101からステップS106までの設定作業が繰り返し行われる。

【0052】以上が、図6のフローチャートの説明であるが、図6のフローチャートのバスリセットからルート決定までの部分と、ルート決定後からID設定終了までの手順をより詳しくフローチャート図に表したものをそれぞれ、図7、図8に示す。

【0053】まず、図7のフローチャートの説明を行う。

【0054】ステップS201としてバスリセットが発生すると、ネットワーク構成は一旦リセットされる。なお、ステップS201としてバスリセットが発生するのを常に監視している。

【0055】次に、ステップS202として、リセットされたネットワークの接続状況を再認識する作業の第一段階として、各機器にリーフ（ノード）であることを示すフラグを立てておく。さらに、ステップS203として各機器が自分の持つポートがいくつ他ノードと接続されているのかを調べる。

【0056】ステップS204のポート数の結果に応じて、これから親子関係の宣言を始めていくために、未定

(7)

11

義（親子関係が決定されていない）ポートの数を調べる。バスリセットの直後はポート数＝未定義ポート数であるが、親子関係が決定されていくにしたがって、ステップS204で検知する未定義ポートの数は変化していくものである。

【0057】まず、バスリセットの直後、はじめに親子関係の宣言を行えるのはリーフに限られている。リーフであるというのはステップS203のポート数の確認で知ることができる。リーフは、ステップS205として、自分に接続されているノードに対して、「自分は子、相手は親」と宣言し動作を終了する。

【0058】ステップS203でポート数が複数ありブランチと認識したノードは、バスリセットの直後はステップS204で未定義ポート数>1ということなので、ステップS206へと移り、まずブランチというフラグが立てられ、ステップS207でリーフからの親子関係宣言で「親」の受付をするために待つ。

【0059】リーフが親子関係の宣言を行い、ステップS207でそれを受けたブランチは適宜ステップS204の未定義ポート数の確認を行い、未定義ポート数が1になっていれば残っているポートに接続されているノードに対して、ステップS205の「自分が子」の宣言をすることが可能になる。2度目以降、ステップS204で未定義ポート数を確認しても2以上あるブランチに対しては、再度ステップS207でリーフ又は他のブランチからの「親」の受付をするために待つ。

【0060】最終的に、いずれか1つのブランチ、又は例外的にリーフ（子宣言を行えるのにすばやく動作しなかった為）がステップS204の未定義ポート数の結果としてゼロになったら、これにてネットワーク全体の親子関係の宣言が終了したものであり、未定義ポート数がゼロ（すべて親のポートとして決定）になった唯一のノードはステップS208としてルートのフラグが立てられ、ステップS209としてルートとしての認識がなされる。

【0061】このようにして、図7に示したバスリセットから、ネットワーク内すべてのノード間における親子関係の宣言までが終了する。

【0062】つぎに、図8のフローチャートについて説明する。

【0063】まず、図8までのシーケンスでリーフ、ブランチ、ルートという各ノードのフラグの情報が設定されているので、これを元にして、ステップS301でそれぞれ分類する。

【0064】各ノードにIDを与える作業として、最初にIDの設定を行うことができるのはリーフからである。リーフ→ブランチ→ルートの順で若い番号（ノード番号＝0～）からIDの設定がなされていく。

【0065】ステップS302としてネットワーク内に存在するリーフの数N（Nは自然数）を設定する。

12

【0066】この後、ステップS303として各自リーフがルートに対して、IDを与えるように要求する。この要求が複数ある場合には、ルートはステップS304としてアービトレーションを行い、ステップS305として勝ったノード1つにID番号を与え、負けたノードには失敗の結果通知を行う。

【0067】ステップS306としてID取得が失敗に終わったリーフは、再度ID要求を出し、同様の作業を繰り返す。IDを取得できたリーフからステップS307として、そのノードのID情報をブロードキャストで全ノードに転送する。1ノードID情報のブロードキャストが終わると、ステップS308として残りのリーフの数が1つ減らされる。

【0068】ここで、ステップS309として、この残りのリーフの数が1以上ある時はステップS303のID要求の作業からを繰り返し行い、最終的にすべてのリーフがID情報をブロードキャストすると、ステップS309がN=0となり、次はブランチのID設定に移る。ブランチのID設定もリーフの時と同様に行われる。

【0069】まず、ステップS310としてネットワーク内に存在するブランチの数M（Mは自然数）を設定する。

【0070】この後、ステップS311として各自ブランチがルートに対して、IDを与えるように要求する。これに対してルートは、ステップS312としてアービトレーションを行い、勝ったブランチから順にリーフに与え終わった次の若い番号から与えていく。

【0071】ステップS313として、ルートは要求を出したブランチにID情報又は失敗結果を通知し、ステップS314としてID取得が失敗に終わったブランチは、再度ID要求を出し、同様の作業を繰り返す。

【0072】IDを取得できたブランチからステップS315として、そのノードのID情報をブロードキャストで全ノードに転送する。

【0073】1ノードID情報のブロードキャストが終わると、ステップS316として残りのブランチの数が1つ減らされる。

【0074】ここで、ステップS317として、この残りのブランチの数が1以上ある時はステップS311のID要求の作業からを繰り返し、最終的にすべてのブランチがID情報をブロードキャストするまで行われる。すべてのブランチがノードIDを取得すると、ステップS317はM=0となり、ブランチのID取得モードも終了する。

【0075】ここまで終了すると、最終的にID情報を取得していないノードはルートのみなので、ステップS318として与えていない番号で最も若い番号を自分のID番号と設定し、ステップS319としてルートのID情報をブロードキャストする。

(8)

13

【0076】以上で、図8に示したように、親子関係が決定した後から、すべてのノードのIDが設定されるまでの手順が終了する。

【0077】次に、一例として図9に示した実際のネットワークにおける動作を図9を参照しながら説明する。

【0078】図9の説明として、(ルート)ノードBの下位にはノードAとノードCが直接接続されており、更にノードCの下位にはノードDが直接接続されており、更にノードDの下位にはノードEとノードFが直接接続された階層構造になっている。この、階層構造やルートノード、ノードIDを決定する手順を以下で説明する。

【0079】バスリセットがされた後、まず各ノードの接続状況を認識するために、各ノードの直接接続されているポート間において、親子関係の宣言がなされる。この親子とは親側が階層構造で上位となり、子側が下位となるということができる。

【0080】図9ではバスリセットの後、最初に親子関係の宣言を行なったのはノードAである。基本的にノードの1つのポートにのみ接続があるノード(リーフと呼ぶ)から親子関係の宣言を行なうことができる。これは自分には1ポートの接続のみということを知ることができるので、これによってネットワークの端であることを認識し、その中で早く動作を行なったノードから親子関係が決定されていく。こうして親子関係の宣言を行なった側(A-B間ではノードA)のポートが子と設定され、相手側(ノードB)のポートが親と設定される。こうして、ノードA-B間では子-親、ノードE-D間で子-親、ノードF-D間で子-親と決定される。

【0081】さらに1階層あがって、今度は複数個接続ポートを持つノード(ブランチと呼ぶ)のうち、他ノードからの親子関係の宣言を受けたものから順次、更に上位に親子関係の宣言を行なっていく。図12ではまずノードDがD-E間、D-F間と親子関係が決定した後、ノードCに対する親子関係の宣言を行っており、その結果ノードD-C間で子-親と決定している。

【0082】ノードDからの親子関係の宣言を受けたノードCは、もう一つのポートに接続されているノードBに対して親子関係の宣言を行なっている。これによってノードC-B間で子-親と決定している。

【0083】このようにして、図9のような階層構造が構成され、最終的に接続されているすべてのポートにおいて親となったノードBが、ルートノードと決定された。ルートは1つのネットワーク構成中に一つしか存在しないものである。

【0084】なお、この図9においてノードBがルートノードと決定されたが、これはノードAから親子関係宣言を受けたノードBが、他のノードに対して親子関係宣言を早いタイミングで行なっていれば、ルートノードは他ノードに移っていたこともあり得る。すなわち、伝達されるタイミングによってはどのノードもルートノード

14

となる可能性があり、同じネットワーク構成でもルートノードは一定とは限らない。

【0085】ルートノードが決定すると、次は各ノードIDを決定するモードに入る。ここではすべてのノードが、決定した自分のノードIDを他のすべてのノードに通知する(ブロードキャスト機能)。

【0086】自己ID情報は、自分のノード番号、接続されている位置の情報、持っているポートの数、接続のあるポートの数、各ポートの親子関係の情報等を含んでいる。

【0087】ノードID番号の割り振りの手順としては、まず1つのポートにのみ接続があるノード(リーフ)から起動することができ、この中から順にノード番号=0、1、2、・・・と割り当てられる。

【0088】ノードIDを手にしたノードは、ノード番号を含む情報をブロードキャストで各ノードに送信する。これによって、そのID番号は「割り当て済み」であることが認識される。

【0089】すべてのリーフが自己ノードIDを取得し終ると、次はブランチへ移りリーフに引き続いたノードID番号が各ノードに割り当てられる。リーフと同様に、ノードID番号が割り当てられたブランチから順次ノードID情報をブロードキャストし、最後にルートノードが自己ID情報をブロードキャストする。すなわち、常にルートは最大のノードID番号を所有するものである。

【0090】以上のようにして、階層構造全体のノードIDの割り当てが終わり、ネットワーク構成が再構築され、バスの初期化作業が完了する。

【0091】アービトレーションについて説明する。

【0092】1394シリアルバスでは、データ転送に先立って必ずバス使用権のアービトレーション(調停)を行なう。1394シリアルバスは個別に接続された各機器が、転送された信号をそれぞれ中継することによって、ネットワーク内すべての機器に同信号を伝えるように、論理的なバス型ネットワークであるので、パケットの衝突を防ぐ意味でアービトレーションは必要である。これによってある時間には、たった一つのノードのみ転送を行なうことができる。

【0093】アービトレーションを説明するための図として図10(a)にバス使用要求の図(b)にバス使用許可の図を示し、以下これを用いて説明する。

【0094】アービトレーションが始まると、1つもしくは複数のノードが親ノードに向かって、それぞれバス使用権の要求を発する。図10(a)のノードCとノードFがバス使用権の要求を発しているノードである。これを受けた親ノード(図10ではノードA)は更に親ノードに向かって、バス使用権の要求を発する(中継する)。この要求は最終的に調停を行なうルートに届けられる。

(9)

15

【0095】バス使用要求を受けたルートノードは、どのノードにバスを使用させるかを定める。この調停作業はルートノードのみが行なえるものであり、調停によって勝ったノードにはバスの使用許可を与える。図10

(b)ではノードCに使用許可が与えられ、ノードFの使用は拒否された図である。

【0096】アービトレーションに負けたノードに対してはDP (data prefix) パケットを送り、拒否されたことを知らせる。拒否されたノードのバス使用要求は次のアービトレーションまで待たされる。

【0097】以上のようにして、アービトレーションに勝ってバスの使用許可を得たノードは、以降データの転送を開始できる。

【0098】ここで、アービトレーションの一連の流れをフローチャート図11に示して、説明する。

【0099】ノードがデータ転送を開始できるためには、バスがアイドル状態であることが必要である。先に行われていたデータ転送が終了して、現在バスが空き状態であることを認識するためには、各転送モードで個別に設定されている所定のアイドル時間ギャップ長（例えば、サブアクション・ギャップ）を経過する事によって、各ノードは自分の転送を開始できると判断する。

【0100】ステップS401として、Asynchデータ、Isoデータ等それぞれ転送するデータに応じた所定のギャップ長が得られたか判断する。所定のギャップ長が得られない限り、転送を開始するために必要なバス使用権の要求はできないので、所定のギャップ長が得られるまで待つ。

【0101】ステップS401で所定のギャップ長が得られたら、ステップS402として転送すべきデータがあるか判断し、ある場合はステップS403として転送するためにバスを確保するよう、バス使用権の要求をルートに対して発する。このときの、バス使用権の要求を表す信号の伝達は、図13に示したように、ネットワーク内各機器を中継しながら、最終的にルートに届けられる。ステップS402で転送するデータがない場合は、そのまま待機する。

【0102】次に、ステップS404として、ステップS403のバス使用要求を1つ以上ルートが受信したら、ルートはステップS405として使用要求を出したノードの数を調べる。

【0103】ステップS405での選択値がノード数=1（使用権要求を出したノードは1つ）だったら、そのノードに直後のバス使用許可が与えられることとなる。ステップS405での選択値がノード数>1（使用要求を出したノードは複数）だったら、ルートはステップS406として使用許可を与えるノードを1つに決定する調停作業を行う。この調停作業は公平なものであり、毎回同じノードばかりが許可を得る様なことはなく、平等に権利を与えていくような構成となっている（フェア・

16

アービトレーション）。

【0104】ステップS407として、ステップS406で使用要求を出した複数ノードの中からルートが調停して使用許可を得た1つのノードと、敗れたその他のノードに分ける選択を行う。ここで、調停されて使用許可を得た1つのノード、またはステップS405の選択値から使用要求ノード数=1で調停無しに使用許可を得たノードには、ステップS408として、ルートはそのノードに対して許可信号を送る。

【0105】許可信号を得たノードは、受け取った直後に転送すべきデータ（パケット）を転送開始する。また、ステップS406の調停で敗れて、バス使用が許可されなかったノードにはステップS409としてルートから、アービトレーション失敗を示すDP (data prefix) パケットを送られ、これを受け取ったノードは再度転送を行うためのバス使用要求を出すため、ステップS401まで戻り、所定ギャップ長が得られるまで待機する。

【0106】以上がアービトレーションの流れを説明した、フローチャート図11の説明である。

【0107】Asynchronous（非同期）転送について説明する。

【0108】アシンクロナス転送は、非同期転送である。図12にアシンクロナス転送における時間的な遷移状態を示す。

【0109】図12の最初のサブアクション・ギャップは、バスのアイドル状態を示すものである。このアイドル時間が一定値になった時点で、転送を希望するノードはバスが使用できると判断して、バス獲得のためのアービトレーションを実行する。

【0110】アービトレーションでバスの使用許可を得ると、次にデータの転送がパケット形式で実行される。データ転送後、受信したノードは転送されたデータに対しての受信結果のask（受信確認用返送コード）をask gapという短いギャップの後、返送して応答するか、応答パケットを送ることによって転送が完了する。askは4ビットの情報と4ビットのチェックサムからなり、成功か、ビジー状態か、ペンディング状態であるかといった情報を含み、すぐに送信元ノードに返送される。

【0111】つぎに、図13にアシンクロナス転送のパケットフォーマットの例を示す。

【0112】パケットには、データ部及び誤り訂正用のデータCRCの他にはヘッダ部があり、そのヘッダ部には図13に示したような、目的ノードID、ソースノードID、転送データ長さや各種コードなどが書き込まれ、転送が行なわれる。

【0113】また、アシンクロナス転送は自己ノードから相手ノードへの1対1の通信である。転送元ノードから転送されたパケットは、ネットワーク中の各ノードに

(10)

17

行き渡るが、自分宛てのアドレス以外のものは無視されるので、宛先の1つのノードのみが読込むことになる。

【0114】以上がアシンクロナス転送の説明である。

【0115】Isochronous（同期）転送について説明する。

【0116】アイソクロナス転送は同期転送である。1394シリアルバスの最大の特徴であるともいえるこのアイソクロナス転送は、特に映像データや音声データといったマルチメディアデータなど、リアルタイムな転送を必要とするデータの転送に適した転送モードである。

【0117】また、アシンクロナス転送（非同期）が1対1の転送であったのに対し、このアイソクロナス転送はブロードキャスト機能によって、転送元の1つのノードから他のすべてのノードへ様に転送される。

【0118】図14はアイソクロナス転送における、時間的な遷移状態を示す図である。

【0119】アイソクロナス転送は、バス上一定時間毎に実行される。この時間間隔をアイソクロナスサイクルと呼ぶ。アイソクロナスサイクル時間は、125 μ Sである。この各サイクルの開始時間を示し、各ノードの時間調整を行なう役割を担っているのがサイクル・スタート・パケットである。サイクル・スタート・パケットを送信するのは、サイクル・マスタと呼ばれるノードであり、1つ前のサイクル内の転送終了後、所定のアイドル期間（サブアクションギャップ）を経た後、本サイクルの開始を告げるサイクル・スタート・パケットを送信する。

【0120】このサイクル・スタート・パケットの送信される時間間隔が125 μ Sとなる。

【0121】また、図14にチャンネルA、チャンネルB、チャンネルCと示したように、1サイクル内において複数種のパケットがチャンネルIDをそれぞれ与えられることによって、区別して転送できる。これによって同時に複数ノード間でのリアルタイムな転送が可能であり、また受信するノードでは自分が欲しいチャンネルIDのデータのみを取り込む。このチャンネルIDは送信先のアドレスを表すものではなく、データに対する論理的な番号を与えているに過ぎない。よって、あるパケットの送信は1つの送信元ノードから他のすべてのノードに行き渡る、ブロードキャストで転送されることになる。

【0122】アイソクロナス転送のパケット送信に先立って、アシンクロナス転送同様アービトレーションが行われる。しかし、アシンクロナス転送のように1対1の通信ではないので、アイソクロナス転送にはask（受信確認用返信コード）は存在しない。

【0123】また、図14に示したisogap（アイソクロナスギャップ）とは、アイソクロナス転送を行なう前にバスが空き状態であると認識するために必要なアイドル期間を表している。この所定のアイドル期間を経過すると、アイソクロナス転送を行ないたいノードは

18

バスが空いていると判断し、転送前のアービトレーションを行なうことができる。

【0124】つぎに、図15にアイソクロナス転送のパケットフォーマットの例を示し、説明する。

【0125】各チャンネルに分かれた、各種のパケットにはそれぞれデータ部及び誤り訂正用のデータCRCの他にヘッダ部があり、そのヘッダ部には図15に示したような、転送データ長やチャンネルNO、その他各種コード及び誤り訂正用のヘッダCRCなどが書き込まれ、転送が行なわれる。

【0126】以上がアイソクロナス転送の説明である。

【0127】バス・サイクルについて説明する。

【0128】実際の1394シリアルバス上の転送では、アイソクロナス転送と、アシンクロナス転送は混在できる。その時の、アイソクロナス転送とアシンクロナス転送が混在した、バス上の転送状態の時間的な遷移の様子を表した図を図16に示す。

【0129】アイソクロナス転送はアシンクロナス転送より優先して実行される。その理由は、サイクル・スタート・パケットの後、アシンクロナス転送を起動するために必要なアイドル期間のギャップ長（サブアクションギャップ）よりも短いギャップ長（アイソクロナスギャップ）で、アイソクロナス転送を起動できるからである。したがって、アシンクロナス転送より、アイソクロナス転送は優先して実行されることとなる。

【0130】図16に示した、一般的なバスサイクルにおいて、サイクル#mのスタート時にサイクル・スタート・パケットがサイクル・マスタから各ノードに転送される。これによって、各ノードで時刻調整を行ない、所定のアイドル期間（アイソクロナスギャップ）を待ってからアイソクロナス転送を行なうべきノードはアービトレーションを行い、パケット転送に入る。図16ではチャンネルeとチャンネルsとチャンネルkが順にアイソクロナス転送されている。

【0131】このアービトレーションからパケット転送までの動作を、与えられているチャンネル分繰り返した後、サイクル#mにおけるアイソクロナス転送がすべて終了したら、アシンクロナス転送を行うことができるようになる。

【0132】アイドル時間がアシンクロナス転送が可能なサブアクションギャップに達する事によって、アシンクロナス転送を行いたいノードはアービトレーションの実行に移れると判断する。

【0133】ただし、アシンクロナス転送が行える期間は、アイソクロナス転送終了後から、次のサイクル・スタート・パケットを転送すべき時間（cycle synch）までの間にアシンクロナス転送を起動するためのサブアクションギャップが得られた場合に限っている。

【0134】図16のサイクル#mでは3つのチャンネル分のアイソクロナス転送と、その後アシンクロナス転送

(11)

19

(含むpack)が2パケット(パケット1、パケット2)転送されている。このアシンクロナスパケット2の後、サイクル $m+1$ をスタートすべき時間(cycle synch)にいたるので、サイクル# m での転送はここまでで終わる。

【0135】ただし、非同期または同期転送動作中に次のサイクル・スタート・パケットを送信すべき時間(cycle synch)に至ったとしたら、無理に中断せず、その転送が終了した後のアイドル期間を待ってから次サイクルのサイクル・スタート・パケットを送信する。すなわち、1つのサイクルが $125\mu\text{S}$ 以上続いたときは、その分次サイクルは基準の $125\mu\text{S}$ より短縮されたとする。このようにアイソクロナス・サイクルは $125\mu\text{S}$ を基準に超過、短縮し得るものである。

【0136】しかし、アイソクロナス転送はリアルタイム転送を維持するために毎サイクル必要であれば必ず実行され、アシンクロナス転送はサイクル時間が短縮されたことによって次以降のサイクルにまわされることもある。こういった遅延情報も含めて、サイクル・マスタによって管理される。

【0137】そこで、まず、第1の実施の形態について説明する。

【0138】図17は、本発明の特徴を最もよく表す図であり、同図において1394のインターフェイスをLANで用いられるOSIモデルの各層と対比させてみると、OSIモデルの物理層1とデータリンク層2が、1394インターフェイスの下位層4であるフィジカル・リンク層に該当し、それら下位層4の上に存在する上位層3が1394インターフェイスではトランスポートプロトコル層5とプレゼンテーション層6に該当する。また、本発明の特徴となるLOGINプロトコル7は、1394インターフェイスの下位層4とトランスポートプロトコル5との間で動作するものである。

【0139】この実施の形態では、シリアルバスプロトコル(SBP-2)8に準拠したデバイスにLOGINプロトコルを挿入することによって、相手のデバイスに対して自分がSBP-2に準拠したプロトコルを使ってやり取りを行いたいことを通知することができる。また例2では、1394インターフェイス上で特化されたデバイスプロトコル9についてもLOGINプロトコルを挿入することで、お互いのデバイスがそのプロトコルがサポートされているかを判別してデータのやり取りを行うことができる。

【0140】図18は、LOGINプロトコルの基本動作を示した図であり、プリンタは印字タスク10を実行する際にまず初めにLOGINプロトコルを使ってプリンタで用意しているプリンタプロトコルA・B・Cのうち、どれを選択して印字するかを決定し、その後は決定されたプロトコルに沿って印字動作を行う。すなわち、プリンタ側でいくつかのプリンタプロトコルをサポート

20

しているデバイスにおいて、ターゲットとの接続の際にまず初めに相手のデバイスのプロトコルをLOGINプロトコルを使って判別し、プリンタは相手のプロトコルに合わせたプリンタプロトコルを複数の中からひとつ選択し、この選ばれたプロトコルに沿って印字データやコマンドのやり取りを行って印字処理を行う。

【0141】図19は、この実施の形態におけるLOGINプロトコルを実装した1394インターフェイスにおける各デバイスの接続形態を示した図で、複数のプリンタプロトコルに対応したプリンタ11に対してLOGINプロトコルを実装したデバイス(PC12、スキャナ13、VCR14等)が接続された場合に、LOGINプロトコルを使用して相手のトランスポートプロトコルに応じてプリンタ側でプリンタプロトコルを切り替えることにより、各デバイスからの印字タスクを問題なく処理することが可能となる。

【0142】図20はログイン動作の流れを示したものである。

【0143】第一ステップ:

20 ・デバイス(この場合マルチプロトコルプリンタ)をロック

・プリンタのケーパビリティ(受け付けるプロトコル等)を取得

かかるケーパビリティは、後述するレジスタ503に格納される。

・ホストのケーパビリティをプリンタにセット

第二ステップ:

・第一ステップで決定したプロトコルで、プリントデータの通信

30 第三ステップ:

・プリンタとホストは、コネクションを切断

【0144】図21は、この実施の形態において、LOGINプロトコルのためにデバイスが備える1394シリアルバスのCRCを示し、図中の501はロックレジスタ(lock)、502はプロトコルレジスタ(protocol)、503はケーパビリティレジスタ(capability)を示す。これらのレジスタは、1394シリアルバスのアドレス空間における初期ユニット空間に定められたアドレスに配置される。

40 【0145】ロックレジスタ501は、リソースのロック状態を表し、「0」以外はロック状態で既にログインされていることを表す。

【0146】ケーパビリティレジスタ503は、設定可能なプロトコルを表し、各ビットが各プロトコルに対応している。ビットの値が「1」で、そのプロトコルが設定可能であることを表し、「0」で設定不可能であることを表す。

【0147】プロトコルレジスタ502は、現在設定されているプロトコルを表し、設定されたプロトコルに対応するケーパビリティレジスタ503のビットに相当す

(12)

21

るビットの値が「1」となる。

【0148】また、プロトコルレジスタ502には、プロトコルを示すビットの他に、ホストデバイスであることを示すhビット、ターゲットデバイスであることを示すtビット、プロトコルマスタであることを示すpビット、及びマルチログイン対応プロトコルであることを示すmビットがある。

【0149】ケーパビリティレジスタ503には、設定可能なプロトコルを示すビットの他に、ホストデバイスとなることができることを示すhビット、ターゲットデ

バイスとなることができることを示すtビット、プロトコルマスタとなることができることを示すpビット、及びプロトコルレジスタ502のmビットに対応する予約ビットがある。

【0150】ここで、プロトコルマスタとは、1394シリアルバスに接続されたLOGINプロトコルをサポートするデバイスの設定可能なプロトコルを取得し、多数決を取り、各デバイスに対して、優先して使用すべきプロトコルを通知するデバイスである。1394シリアルバス上にプロトコルマスタになることができるデバイスが複数存在する場合には、ノード間で調整することで、プロトコルマスタを決定する。

【0151】図22は、プロトコルマスタが行うプロトコルの多数決処理を示すフローチャートである。

【0152】まず、プロトコルの集計のための初期化処理を行う（ステップS801）。

【0153】次に、全てのノードの情報を取得したかを判定する（ステップS802）。

【0154】ステップS802の判定の結果、全てのノードの集計が終了していない場合には、着目しているノードのケーパビリティレジスタ502の内容を読み出し、設定可能なプロトコルを取得する（ステップS803）。

【0155】そして、設定可能なプロトコル毎に、ノードを集計し（ステップS804）、次のノードの処理のために、ステップS802に戻る。

【0156】全てのノードの集計が終了したら、すなわちステップS802の判定の結果、全てのノードの集計が終了していた場合、通知のための初期化処理を行う（ステップS805）。

【0157】次に、全てのノードに通知したかを判定する（ステップS806）。

【0158】ステップS806の判定の結果、全てのノードへの通知が終了していた場合には、本処理を終了する。

【0159】一方、全てのノードへの通知が終了していない場合には、優先プロトコルを通知し（ステップS807）、次のノードの処理のために、ステップS806に戻る。

【0160】図23は、ホスト側のログイン処理を示し

22

たフローチャートである。

【0161】ログインを開始するためには、まず、ログインしようとするプリンタ（ターゲット）のロックレジスタ501、プロトコルレジスタ502、及びケーパビリティレジスタ503の内容をリードトランザクションにより取得する（ステップS600）。

【0162】次に、ステップS600で得られたケーパビリティレジスタ503の内容から、ホストが通信に用いるプロトコルをプリンタがサポートしている（ログイン処理開始OK）か否かを判別する（ステップS601）。

【0163】ステップS601の判別結果により、ホストが通信に用いるプロトコルをプリンタがサポートしていなかった場合には、本処理を終了する。これにより、このログイン処理が中止される。

【0164】一方、ホストが通信に用いるプロトコルをプリンタがサポートしていた場合、ステップS600で得られたロックレジスタ501の内容から、その値が「0」か否かを判別することで、プリンタのリソースがロック状態で既にログインされているか否かを判別する（ステップS602）。

【0165】ステップS602の判別結果により、プリンタが他のデバイスによりログイン中（ロックレジスタ501≠0）であった場合、ステップS600で得られたプロトコルレジスタ502の内容から、現在設定されているプロトコルがマルチログイン対応か否かを判別する（ステップS608）。

【0166】ステップS608の判別の結果により、マルチログイン対応であった場合には、後述するステップS606に進み、マルチログイン対応でなかった場合には、本処理を終了する。

【0167】ステップS602の判別結果により、プリンタが他のデバイスによりログイン中でなくログイン可能（ロックレジスタ501=0）であった場合、リソースロック処理を行うことで、プリンタのロックレジスタ501にロックトランザクションを用いて「1」を書き込み、ホスト側のログイン設定とする（ステップS603）。このステップS603により、プリンタはロックされたこととなり、他のデバイスからのレジスタ内容の変更は不可能な状態となる。

【0168】プリンタのリソースがロックされた状態になると、次に、プロトコルの設定を行うが、この実施の形態におけるプリンタは、複数のプリンタプロトコルをサポートするため、プリントデータを受けるまえに、ホスト側との間で使用するプロトコルを決定する必要がある。

【0169】そこで、ここでは、ホスト側は、上述したプロトコルマスタから通知されたプロトコルをホスト及びプリンタの双方がサポートすれば、これを優先する。また、ホスト及びプリンタの双方がサポートするプロト

(13)

23

コルにマルチログイン対応のプロトコルがあれば、これを優先する。

【0170】このようにして使用するプロトコルが決定されると、このプロトコルがプリンタに設定されているプロトコルであるか否かを判別する（ステップS604）。

【0171】ステップS604の判別の結果により、決定されたプロトコルとプリンタに設定されているプロトコルが異なっている場合に、プロトコル設定処理を行うことで、プロトコルを変更する（ステップS605）。このプロトコル設定処理は、プリンタのプロトコルレジスタ502に対して、ライトトランザクションにより、これから使用するプロトコルに相当するビットを「1」に設定して通知する、という処理である。

【0172】この時点で、ホストが通信に用いるプロトコルがプリンタに通知され、且つプリンタがロック状態であるため、現在ログインしているホストが通常のプロトコルでプリントデータの送信を行う（ステップS606）。また、通信に用いるプロトコルにログインが必要な場合には、そのプロトコルでログインする。さらに、そのプロトコルがマルチログイン対応の場合も、ここでログインする。

【0173】そして、プリントデータの送信が終了すると、ログアウト処理を行って（ステップS607）、本処理を終了する。すなわち、ホストは、プリンタのロックレジスタ501をクリアすることで、プリンタのリソースロックを解除する。また、マルチログイン対応のプロトコルで通信していた場合には、他のデバイスがログインしていない場合に、プリンタのリソースロックを解除する。

【0174】図24は、プリント側（ターゲット側）のログイン処理を示したフローチャートである。

【0175】プリンタは、通常ホストからのログイン待ち状態にいる（ステップS701）。このとき、ホストからのプリントリクエストは、プリンタのロックレジスタ501、プロトコルレジスタ502、及びケーパビリティレジスタ503の読み取りにより開始されるため、これらのレジスタを、常に他のデバイスからの読出可能状態にしておく。

【0176】そこで、今、プリントアウトを実行しようとするホストが、プリンタのロックレジスタ501へ書き込みを行うことで、プリンタがロックされたとする。

【0177】プリンタがロックされると、まず、ホストからの使用プロトコルの再設定の通知を判別する（ステップS702）。

【0178】ステップS702の判別の結果により、プロトコルの通知があった場合には、プリンタは、自分の受け取るプロトコルをスイッチし、ホスト側に合わせてプロトコルの再設定を行って（ステップS703）、次のステップS704に進む。

24

【0179】ステップS702の判別の結果により、プロトコルの通知がなかった場合には、ステップS703の処理は行わずに、ステップS704に進む。

【0180】次に、ステップS704～ステップS707により、使用プロトコルによる通信を行う。ここで、マルチログイン対応のプロトコルでは、このプロトコルでの複数デバイスによるログインを受け付ける。

【0181】そして、全ての通信が終了すると、プリンタは、ロックレジスタ501がホストによりクリアされたのを確認して、再びログイン待ち状態（ステップS701）に戻る（ステップS708）。

【0182】尚、本発明は、図19に示すような、複数の機器から構成されるシステムに適用しても、1つの機器からなる装置内のデータ処理方法に適用してもよい。

【0183】また、本発明の目的は、上述した各実施の形態のホスト及び端末の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（又はCPUやMPU）が記憶媒体に格納されたプログラムコードを読みだして実行することによっても、達成されることは言うまでもない。

【0184】この場合、記憶媒体から読み出されたプログラムコード自体が前述した各実施の形態の機能を実現することとなり、そのプログラムコードを記憶した記憶媒体は本発明を構成することとなる。

【0185】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROM等を用いることができる。

【0186】また、コンピュータが読みだしたプログラムコードを実行することにより、前述した実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているOS等が実際の処理の一部又は全部を行い、その処理によって実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0187】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された拡張機能ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0188】

【発明の効果】以上説明したように本発明によれば、拡張性の高いデータ処理方法、データ処理装置、プリンタ及び記憶媒体を提供することができる。特に、複数のプロトコルから1つのプロトコルを選択して使用する場合

(14)

25

に、プロトコルの変更の頻度を小さくして、プロトコルの再設定による効率の低下を防ぐことができる。また、複数デバイスのログインを許容するプロトコルを優先することで、デバイスの占有を防ぐことができる。

【図面の簡単な説明】

【図1】IEEE 1394ケーブルを用いた通信システムの一例を示す図である。

【図2】IEEE 1394の階層構造を示す図である。

【図3】IEEE 1394のアドレスを示す図である。

【図4】IEEE 1394のケーブル断面図である。

【図5】DS-Link 符号化方式を説明するための図である。

【図6】バスリセットからIDの設定までを説明するフローチャートである。

【図7】ルートの決定方法を説明するフローチャートである。

【図8】親子関係決定からすべてのノードIDの設定までの手順を説明するフローチャートである。

【図9】ノード間の親子関係を示す図である。

【図10】アービトレーションの過程を説明するための図である。

【図11】アービトレーションの過程を示すフローチャートである。

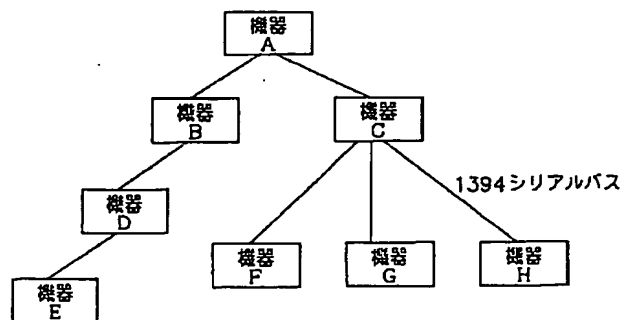
【図12】Asynchronous転送におけるサブアクションを示す図である。

【図13】Asynchronous転送におけるパケット構造を示す図である。

【図14】Isochronous 転送におけるサブアクションを示す図である。

【図15】Isochronous 転送におけるパケット構造を示す図である。

【図1】



26

【図16】IEEE 1394の通信サイクルの一例を説明するための図である。

【図17】本発明を適用したシステムを説明するための図である。

【図18】LOGINプロトコルの基本動作を説明するための図である。

【図19】LOGINプロトコルを実装した1394インターフェイスにおける各デバイスの接続形態を説明するための図である。

【図20】ログイン動作の流れを説明するための図である。

【図21】LOGINプロトコルのためにプリンタが備える1394シリアルバスのCSRを説明するための図である。

【図22】プロトコルマスタのプロトコル多数決処理を説明するためのフローチャートである。

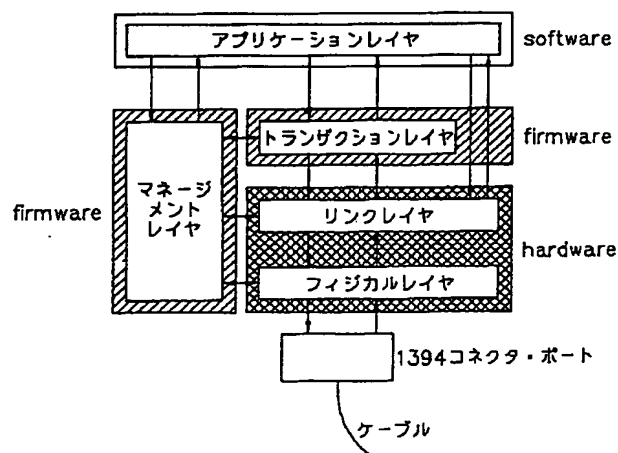
【図23】ログイン処理のホスト側の処理を説明するためのフローチャートである。

【図24】ログイン処理のプリンタ側の処理を説明するためのフローチャートである。

【符号の説明】

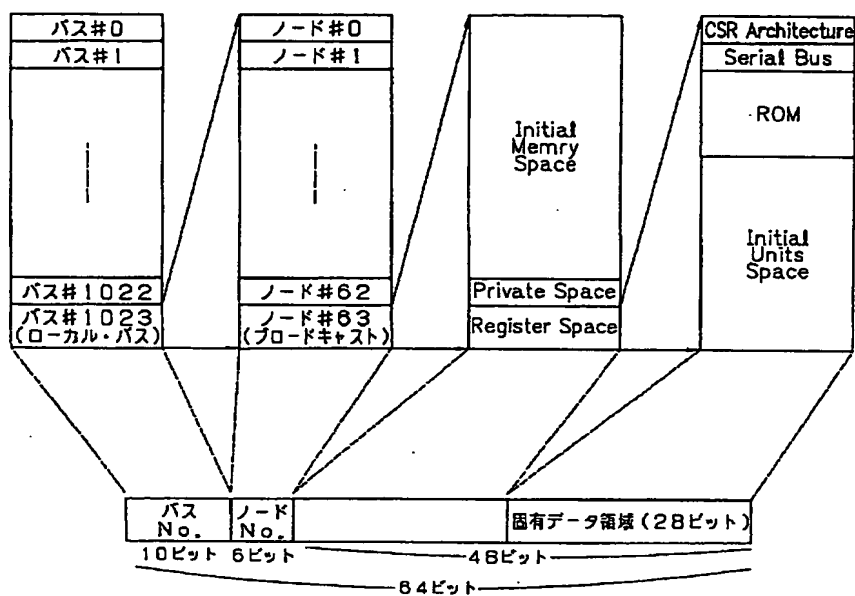
- 1 OSIモデルの物理層
- 2 データリンク層
- 3 上位層
- 4 上位層
- 5 トランスポートプロトコル層
- 6 プレゼンテーション層
- 7 LOGINプロトコル
- 8 シリアルバスプロトコル (SBP-2)
- 9 デバイスプロトコル

【図2】

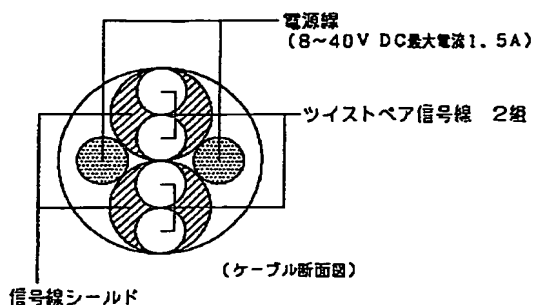


(15)

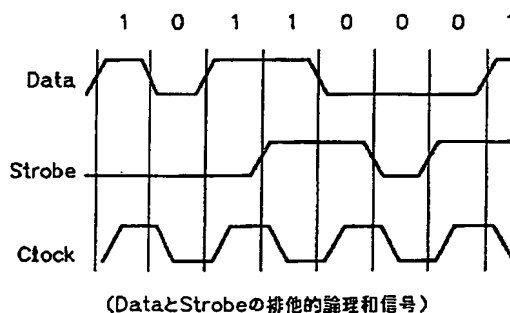
【図 3】



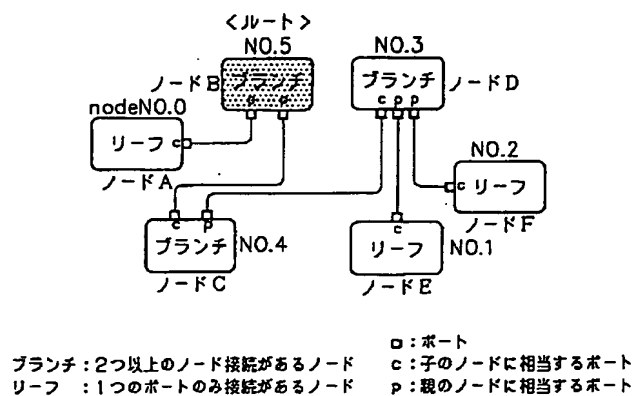
【図 4】



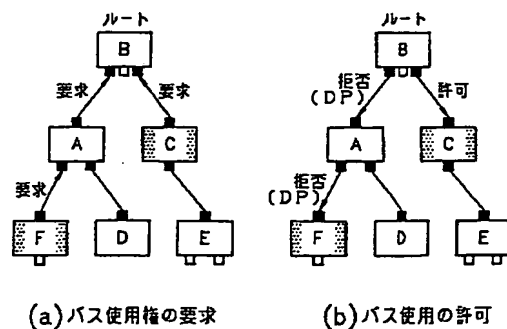
【図 5】



【図 9】

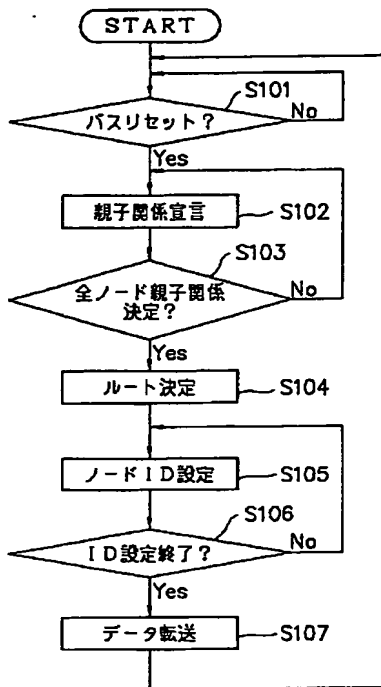


【图 10】

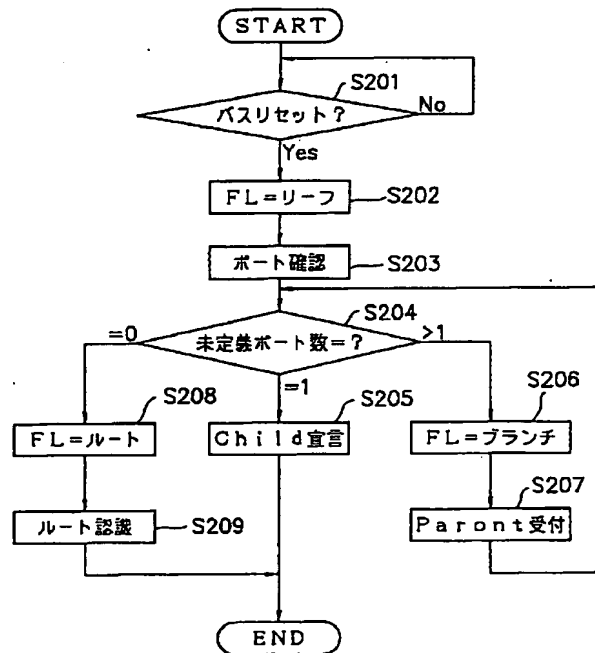


(16)

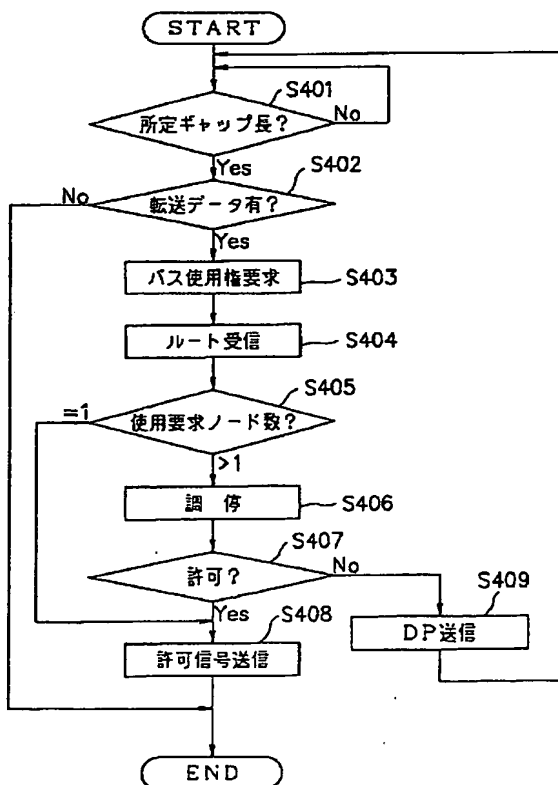
【図6】



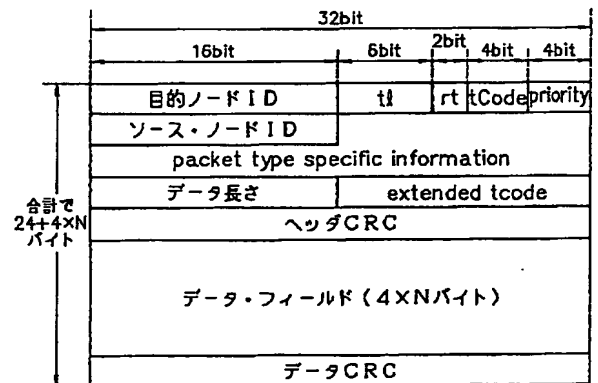
【図7】



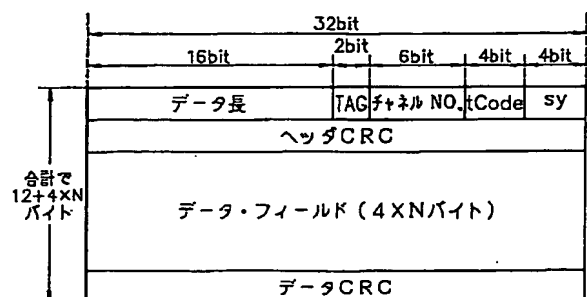
【図11】



【図13】

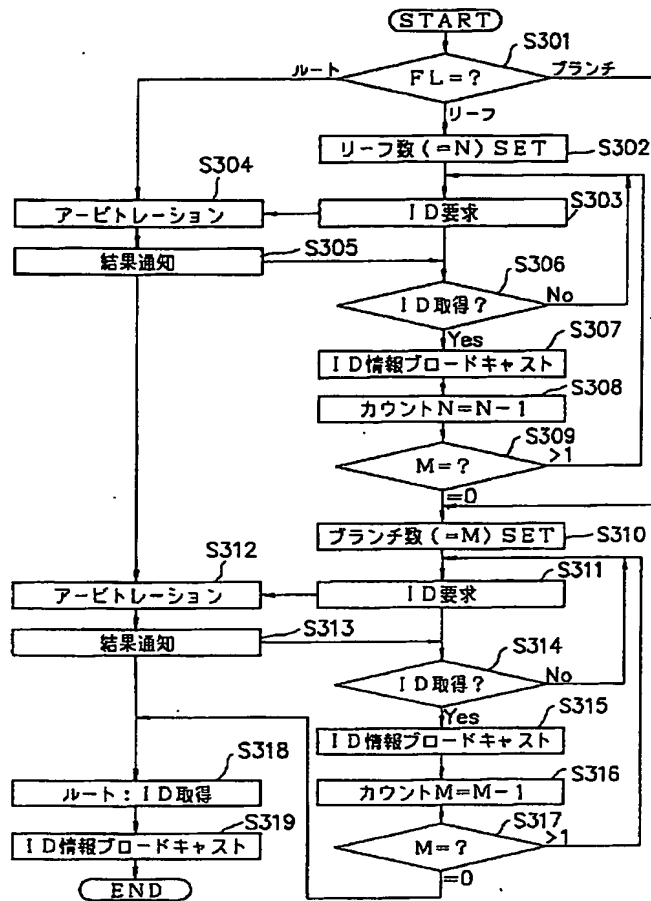


【図15】

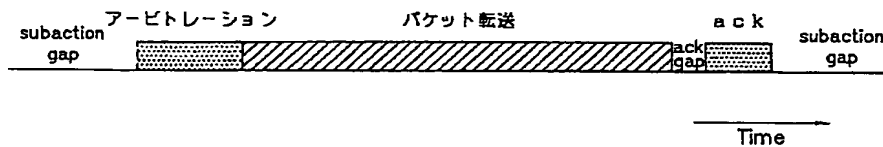


(17)

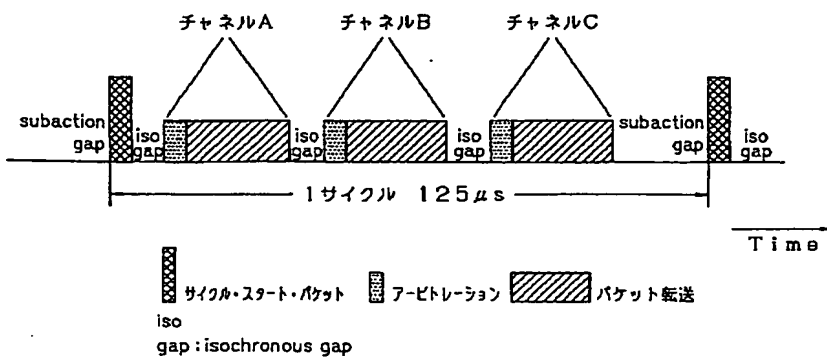
【図 8】



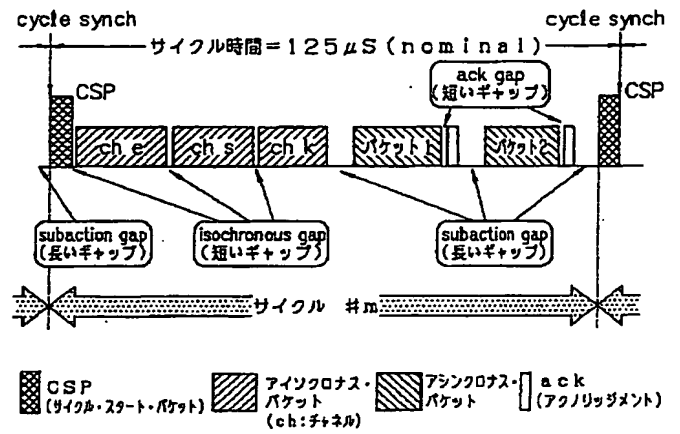
【図 12】



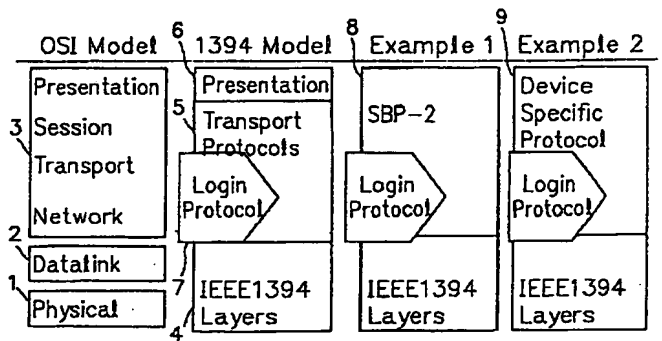
【図 14】



【図 16】

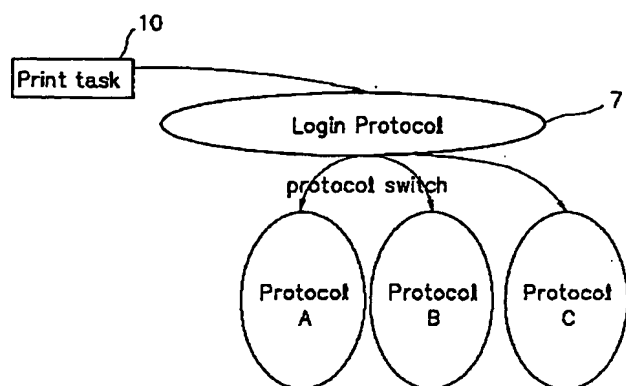


【図 17】

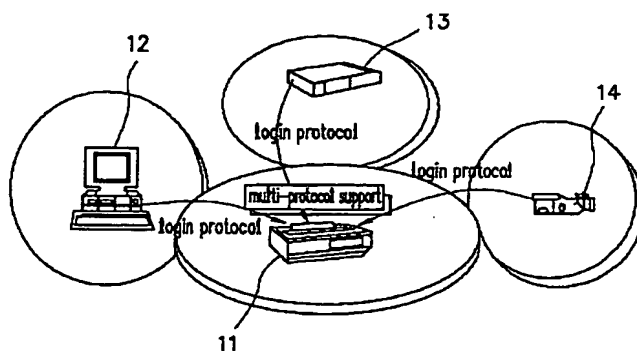


(18)

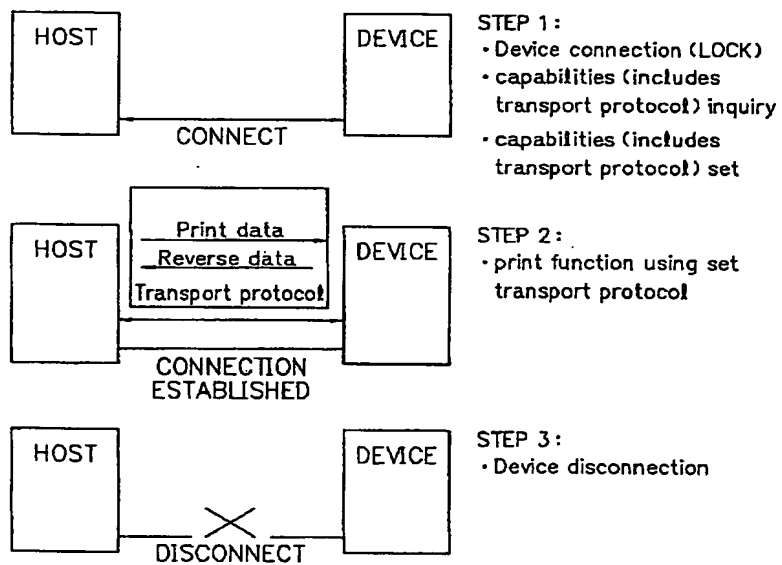
【図18】



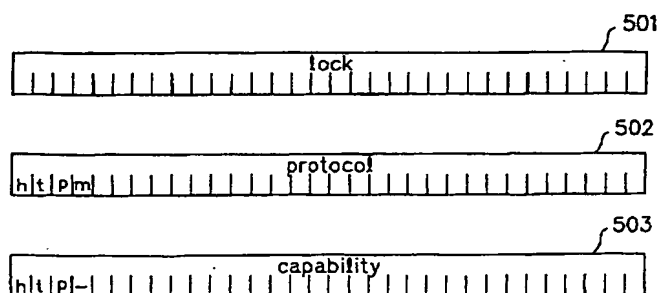
【図19】



【図20】

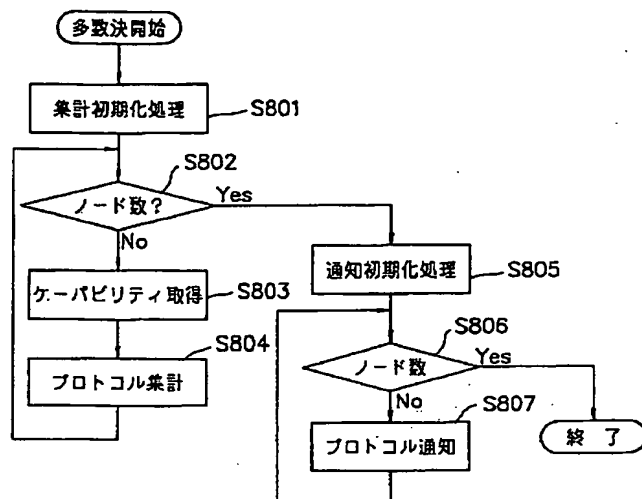


【図21】

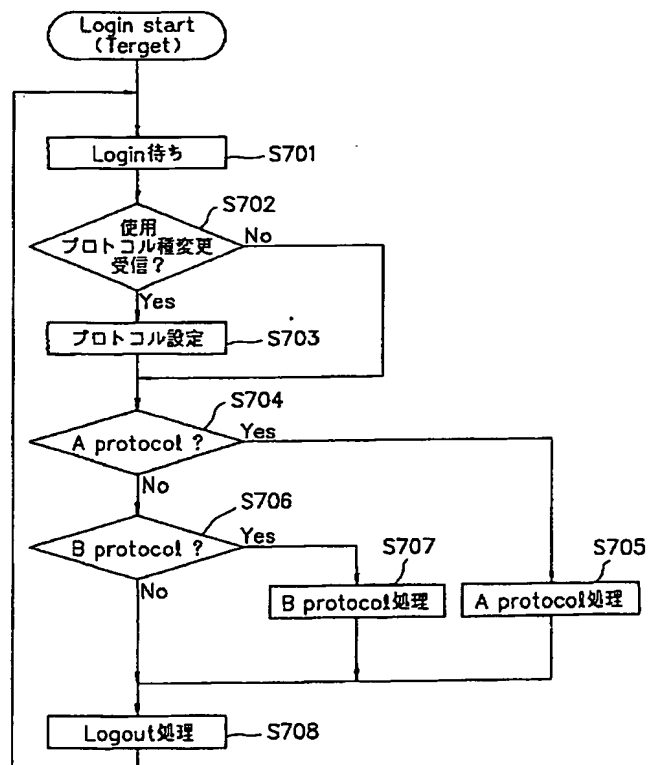


(19)

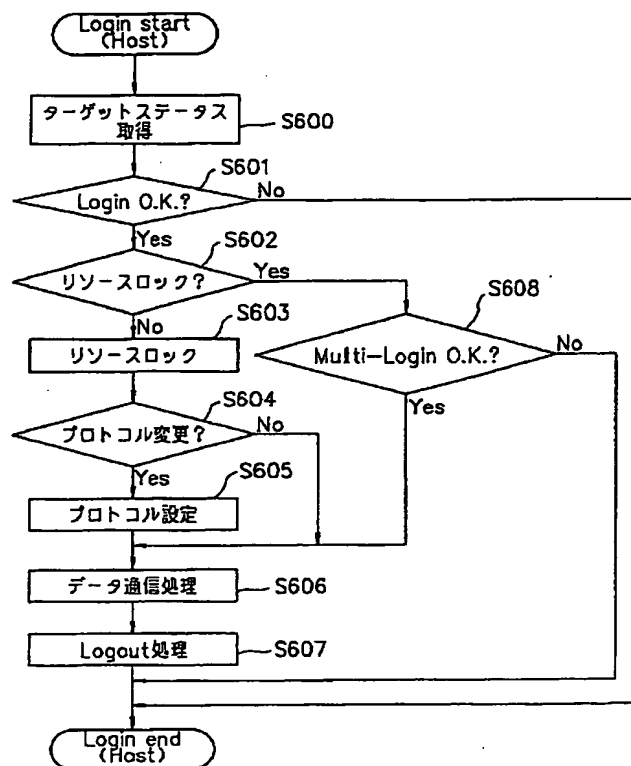
【図22】



【図24】



【図23】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-283136

(43)Date of publication of application : 23.10.1998

(51)Int.Cl.

G06F 3/12
B41J 29/38
// G06F 13/00

(21)Application number : 09-089826

(71)Applicant : CANON INC

(22)Date of filing : 08.04.1997

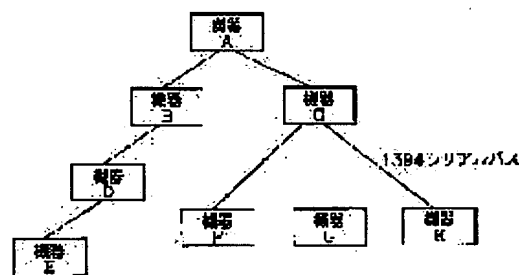
(72)Inventor : KATANO KIYOSHI
NAKAMURA ATSUSHI

(54) DATA PROCESSING METHOD, DATA PROCESSOR, PRINTER AND STORAGE MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a data processing method with high extendibility at the time of sending out data through a serial bus by giving a specified protocol priority at the time of selecting a protocol to be executed from the plural kinds of the protocols following an initial protocol.

SOLUTION: Equipments A-H are provided and A and B, A and C, B and D, D and E, C and F, C and G and C and H are respectively connected by the twist pair cable of a 1394 serial bus. The respective equipments A-H are respectively provided with intrinsic IDs. In such a system, the initial protocol unrelated to the kind of the protocol is executed, and at the time of selecting the protocol to be executed from the plural kinds of the protocols following the initial protocol, the specified protocol is given priority. In this case, the specified protocol is the protocol of the largest number of supporting devices in the devices connected onto a common serial bus.



LEGAL STATUS

[Date of request for examination]

28.12.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3495879

[Date of registration]

21.11.2003

[Number of appeal against examiner's decision of rejection]

*** NOTICES ***

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The data-processing approach characterized by giving priority to a specific protocol in case it is the data-processing approach for switching two or more kinds of protocols through a serious-in common bus, the initial protocol without regards to the class of said protocol is performed and said protocol which should be performed from the protocol of a class is chosen following said initial protocol.

[Claim 2] Said specific protocol is the data-processing approach according to claim 1 characterized by being most protocols of the device supported among the devices connected on said common serial bus.

[Claim 3] The device of the arbitration of the devices connected on said common serial bus is the data-processing approach according to claim 2 which detects the number of the devices which support the protocol on said common serial bus, and is characterized by notifying most protocols of the device to support to each device as said specific protocol for every protocol.

[Claim 4] Said specific protocol is the data-processing approach according to claim 1 characterized by being the protocol which permits the log in of two or more devices.

[Claim 5] Said common serial bus is the data-processing approach according to claim 1 characterized by being the bus which suited IEEE1394 specification.

[Claim 6] Said common serial bus is the data-processing approach according to claim 1 characterized by being the bus which suited USB specification.

[Claim 7] Said initial protocol is the data-processing approach according to claim 1 characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model.

[Claim 8] The data processor which is a data processor for switching two or more kinds of protocols through a serious-in common bus, and is characterized by having the 1st activation means which performs the initial protocol without regards to the class of said protocol, and the 2nd activation means which gives priority to a specific protocol in case said protocol which should be performed from the protocol of a class is chosen following said initial protocol.

[Claim 9] Said 2nd activation means is a data processor according to claim 8 characterized by making into said specific protocol most protocols of the device supported among the devices connected on said common serial bus.

[Claim 10] Said 2nd activation means is a data processor according to claim 9 which detects the number of the devices which support the protocol on said common serial bus, and is characterized by notifying most protocols of the device to support to each device as said specific protocol for every protocol with the device of the arbitration of the devices connected on said common serial bus.

[Claim 11] Said 2nd activation means is a data processor according to claim 8 characterized by making into said specific protocol the protocol which permits the log in of two or more devices.

[Claim 12] Said common serial bus is a data processor according to claim 8 characterized by being the bus which suited IEEE1394 specification.

[Claim 13] Said common serial bus is a data processor according to claim 8 characterized by being the bus which suited USB specification.

[Claim 14] Said initial protocol is a data processor according to claim 8 characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model.

[Claim 15] The printer which is a printer which switches two or more kinds of protocols through a serious-in common bus, and is characterized by having the 1st activation means which performs the initial protocol

without regards to the class of said protocol, and the 2nd activation means which gives priority to a specific protocol in case said protocol which should be performed from the protocol of a class is chosen following said initial protocol.

[Claim 16] Said 2nd activation means is a printer according to claim 15 characterized by making into said specific protocol most protocols of the device supported among the devices connected on said common serial bus.

[Claim 17] Said 2nd activation means is a printer according to claim 16 which detects the number of the devices which support the protocol on said common serial bus, and is characterized by notifying most protocols of the device to support to each device as said specific protocol for every protocol with the device of the arbitration of the devices connected on said common serial bus.

[Claim 18] Said 2nd activation means is a printer according to claim 15 characterized by making into said specific protocol the protocol which permits the log in of two or more devices.

[Claim 19] Said common serial bus is a printer according to claim 15 characterized by being the bus which suited IEEE1394 specification.

[Claim 20] Said protocol is a printer according to claim 15 characterized by being a protocol for ink jet printers.

[Claim 21] Said common serial bus is a printer according to claim 15 characterized by being the bus which suited USB specification.

[Claim 22] Said initial protocol is a printer according to claim 15 characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model.

[Claim 23] The printer according to claim 15 characterized by having a transmission means to transmit the image data which should be printed after protocol activation of said arbitration.

[Claim 24] The printer according to claim 23 characterized by having a photography means to obtain said image data by photo electric conversion.

[Claim 25] The storage characterized by to store the 1st execute step which performs the initial protocol without regards to [are the storage with which the step for performing data processing for switching two or more kinds of protocols through a serious-in common bus was stored, and] the class of said protocol, and the 2nd execute step which gives priority to a specific protocol in case said protocol which should be performed from the protocol of a class is chosen following said initial protocol possible [read-out of a computer].

[Claim 26] Said 2nd execute step is a storage according to claim 25 characterized by including the step which makes most protocols of the device supported among the devices connected on said common serial bus said specific protocol.

[Claim 27] Said 2nd execute step is a storage according to claim 26 which detects the number of the devices which support the protocol on said common serial bus, and is characterized by including the step which notifies most protocols of the device to support to each device as said specific protocol for every protocol with the device of the arbitration of the devices connected on said common serial bus.

[Claim 28] Said 2nd execute step is a printer according to claim 25 characterized by including the step which makes the protocol which permits the log in of two or more devices said specific protocol.

[Claim 29] Said common serial bus is a storage according to claim 25 characterized by being the bus which suited IEEE1394 specification.

[Claim 30] Said common serial bus is a storage according to claim 25 characterized by being the bus which suited USB specification.

[Claim 31] Said initial protocol is a storage according to claim 25 characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model.

[Translation done.]

*** NOTICES ***

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the data-processing approach, the data processor, printer, and storage for switching two or more kinds of protocols through a serious-in common bus.

[0002]

[Description of the Prior Art] Conventionally, the system of various classes is known as a system which sends out data to a printer through a serious bus.

[0003] For example, the technique which outputs data to a printer from a computer is known using the interface of a de facto standard with which SCSI (Small Computer System Interface), Centronics, etc. generally came to be used widely.

[0004]

[Problem(s) to be Solved by the Invention] However, the conventional protocol which sends out data using a serious bus was restricted to one of the manufacturer propers. Therefore, the problem that expandability was missing had arisen. Especially the trouble that this expandability was missing in case data are outputted using the interface which connects the device of various classes, for example, an interface like IEEE1394, was the big technical problem which should be solved.

[0005] Then, in case this invention was accomplished in order to remove the above-mentioned fault, and it sends out data through sill ARUBASU, it aims at offering the high data-processing approach, the data processor, printer, and storage of expandability.

[0006]

[Means for Solving the Problem] This invention is the data-processing approach for switching two or more kinds of protocols through a serious-in common bus, and in case it performs the initial protocol without regards to the class of said protocol and chooses said protocol which should be performed from the protocol of a class following said initial protocol, it is characterized by giving priority to a specific protocol. Moreover, said specific protocol is characterized by being most protocols of the device supported among the devices connected on said common serial bus. Moreover, for every protocol, the device of the arbitration of the devices connected on said common serial bus detects the number of the devices which support the protocol on said common serial bus, and is characterized by notifying most protocols of the device to support to each device as said specific protocol. Moreover, said specific protocol is characterized by being the protocol which permits the log in of two or more devices. Moreover, said common serial bus is characterized by being the bus which suited IEEE1394 specification. Moreover, said common serial bus is characterized by being the bus which suited USB specification. Moreover, said initial protocol is characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model. This invention is a data processor for switching two or more kinds of protocols through a serious-in common bus, and is characterized by having the 1st activation means which performs the initial protocol without regards to the class of said protocol, and the 2nd activation means which gives priority to a specific protocol in case said protocol which should be performed from the protocol of a class is chosen following said initial protocol. Moreover, said 2nd activation means is characterized by making into said specific protocol most protocols of the device supported among the devices connected on said common serial bus. Moreover, with the device of the arbitration of the devices connected on said common serial bus, for every protocol, said 2nd activation means detects the number of the devices which support the protocol on said common serial bus, and is characterized by notifying most protocols of the device

to support to each device as said specific protocol. Moreover, said 2nd activation means is characterized by making into said specific protocol the protocol which permits the log in of two or more devices. Moreover, said common serial bus is characterized by being the bus which suited IEEE1394 specification. Moreover, said common serial bus is characterized by being the bus which suited USB specification. Moreover, said initial protocol is characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model. This invention is a printer which switches two or more kinds of protocols through a serious-in common bus, and is characterized by having the 1st activation means which performs the initial protocol without regards to the class of said protocol, and the 2nd activation means which gives priority to a specific protocol in case said protocol which should be performed from the protocol of a class is chosen following said initial protocol. Moreover, said 2nd activation means is characterized by making into said specific protocol most protocols of the device supported among the devices connected on said common serial bus. Moreover, with the device of the arbitration of the devices connected on said common serial bus, for every protocol, said 2nd activation means detects the number of the devices which support the protocol on said common serial bus, and is characterized by notifying most protocols of the device to support to each device as said specific protocol. Moreover, said 2nd activation means is characterized by making into said specific protocol the protocol which permits the log in of two or more devices. Moreover, said common serial bus is characterized by being the bus which suited IEEE1394 specification. Moreover, said protocol is characterized by being a protocol for ink jet printers. Moreover, said common serial bus is characterized by being the bus which suited USB specification. Moreover, said initial protocol is characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model. Moreover, it is characterized by having a transmission means to transmit the image data which should be printed after protocol activation of said arbitration. Moreover, it is characterized by having a photography means to obtain said image data by photo electric conversion. This invention is the storage with which the step for performing data processing for switching two or more kinds of protocols through a serious-in common bus was stored. The 1st execute step which performs the initial protocol without regards to the class of said protocol, In case said protocol which should be performed from the protocol of a class is chosen following said initial protocol, it is characterized by storing the 2nd execute step which gives priority to a specific protocol possible [read-out of a computer]. Moreover, said 2nd execute step is characterized by including the step which makes most protocols of the device supported among the devices connected on said common serial bus said specific protocol. Moreover, with the device of the arbitration of the devices connected on said common serial bus, for every protocol, said 2nd execute step detects the number of the devices which support the protocol on said common serial bus, and is characterized by including the step which notifies most protocols of the device to support to each device as said specific protocol. Moreover, said 2nd execute step is characterized by including the step which makes the protocol which permits the log in of two or more devices said specific protocol. Moreover, said common serial bus is characterized by being the bus which suited IEEE1394 specification. Moreover, said common serial bus is characterized by being the bus which suited USB specification. Moreover, said initial protocol is characterized by being the protocol performed in the layer of a high order rather than the data link layer of an OSI model.

[0007]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained using a drawing.

[0008] Here, first, with the gestalt of the 1st explained below and the 2nd operation, as digital I/F which connects between each device, since the IEEE1394 serial bus is used, an IEEE1394 serial bus is explained beforehand.

[0009] With the appearance of noncommercial digital one VCR and a DVD player, it is the real time of a video data, audio data, etc., and the support of the data transfer of high amount of information is needed. In order to transmit such video datas and audio data on real time and to transmit to the digital instrument of downloading to a personal computer (PC) ****, or others, the interface which the interface equipped with the required transfer facility in which a high-speed-data transfer is possible is needed, and was developed from such viewpoints is IEEE1394 -1995 (High Performance Serial Bus) (only henceforth a 1394 serial bus).

[0010] The example of the network system constituted by using a 1394 serial bus for drawing 1 is shown.

[0011] This system is equipped with Devices A, B, C, D, E, F, G, and H, and between A-B, between A-C,

between B-D, between D-E, between C-F, between C-G, and between C-H are connected with the twisted-pair cable of a 1394 serial bus, respectively. This device A-H is a personal computer, digital VTR, DVD, a digital camera, a hard disk, a monitor, a tuner, a monitor, etc.

[0012] The connection type between each device enables mixture of a daisy chain method and node multipoint system, and high connection of a degree of freedom is possible for it.

[0013] Moreover, each device has ID of a proper each one, and when each recognizes each other, it constitutes one network in the range connected by 1394 serial bus. Sequential connection of between each digital instrument is only made with one 1394 serial bus cable, respectively, and each device performs the role of junction and constitutes one network as a whole. Moreover, when a cable is connected to a device by the 1394 serial bus and the Plug & Play function, it has the function to recognize recognition, a connection situation, etc. of a device automatically.

[0014] Moreover, in a system as shown in drawing 1, when a certain device is deleted from a network or it is newly added, after performing bus reset automatically and resetting the network configuration till then, a new network is reconstructed. By this function, the configuration of that occasional network can always be set up and recognized.

[0015] Moreover, it has with 100/200/400Mbps, a device with the transfer rate of a high order supports a low-ranking transfer rate, and a data transfer rate takes transposition.

[0016] As data transfer mode, there are Asynchronous transfer mode which transmits asynchronous datas (Asynchronous data: henceforth Async data), such as a control signal, and Isochronous transfer mode which transmits synchronous data (Isochronous data: following and Iso data ****), such as a real time video data and audio data. After this Async data and Iso data transmit the cycle-start packet (CSP) which shows cycle initiation in each cycle (usually one cycle 125microS), giving priority to an Iso data transfer over Async data, within a cycle, it is intermingled and they are transmitted.

[0017] Below, the component of a 1394 serial bus is shown in drawing 2.

[0018] The 1394 serial bus consists of layer (hierarchy) structures as a whole. As shown in drawing 2, there is a connector port where a connector is connected with the cable of a 1394 serial bus, and the physical layer and the link layer are positioned as hardware on it.

[0019] The hardware section is the part of a substantial interface chip, among those a physical layer performs coding, connector-related control, etc., and a link layer performs packet transfer, control of the cycle time, etc.

[0020] The transaction layer of the firmware section manages the data which should be transmitted (transaction), and issues the instruction of Read, Write, and Lock. A management layer is a part which performs management of the connection situation of each device connected, or ID, and manages a network configuration.

[0021] Even this hardware and firmware are the configuration of the 1394 serial bus on parenchyma.

[0022] Moreover, the application layers of the software section are how data are carried on an interface by changing with software to be used, and a part to specify, and are a printer and AVC. The protocol etc. is specified.

[0023] The above is the configuration of a 1394 serial bus.

[0024] Below, drawing of the address space in a 1394 serial bus is shown in drawing 3.

[0025] 64 bit addresses of each node proper are surely given to each device (node) connected to the 1394 serial bus. And by storing this address in ROM, the node address of itself's or a partner can always be recognized and the communication link which specified the partner can also be performed.

[0026] Addressing of a 1394 serial bus is a method according to IEEE1212 specification, 10 bits of the beginning are used for assignment of the number of a bus, and, as for address selection, the following 6 bits are used for assignment of a node ID number. The remaining 48 bits become the address width of face given to the device, and can use it as an address space of a proper, respectively. 28 bits of the last store discernment of each device, the information on assignment of a service condition, etc. as a field of proper data.

[0027] The above is the outline of the technique of a 1394 serial bus.

[0028] Below, the part of the technique which can be called description of a 1394 serial bus is explained more to a detail.

[0029] The electric specification of a 1394 serial bus is explained.

[0030] The sectional view of a 1394 serial bus cable is shown in drawing 4.

- [0031] By 1394 serial bus, power-source Rhine other than six pins, i.e., 2 sets of twisted pair signal lines, is prepared in the interconnection cable. Supply of power is attained at the device which does not have a power source by this, the device which carried out sag by failure.
- [0032] 8-40V, and a current are specified for the electrical potential difference of the power source which flows the inside of a power-source line as maximum current DC1.5A.
- [0033] In addition, it is constituted from the specification called DV cable by four pins which excluded the power source.
- [0034] DS-Link coding is explained.
- [0035] Drawing for explaining the DS-Link coding method of a data transfer format adopted by 1394 serial bus is shown in drawing 5.
- [0036] By 1394 serial bus, the DS-Link (Data/Strobe Link) coding method is adopted. This DS-Link coding method is suitable for the high-speed serial data communication link, and that configuration needs two signal lines. It has the composition of sending a strobe signal to the twisted pair line from that of delivery and another side in the data which become main one in a twisted pair wire. In a receiving side, a clock is reproduced by taking the exclusive OR of this data that communicates, and a strobe.
- [0037] Since there is no need of sending the information which shows that it is an idle state when there are not the circuit scale of Controller LSI being made small and data which should be transmitted further, since transfer efficiency's being high and a PLL circuit become unnecessary as a merit using this DS-Link coding method compared with 8 / 10B conversion, it is mentioned by the ability making the transceiver circuit of each device into sleeping that reduction of power consumption can be aimed at etc.
- [0038] The sequence of bus reset is explained.
- [0039] By 1394 serial bus, Node ID is given to each device (node) connected, and it is recognized as network configuration.
- [0040] When this network configuration has change (for example, when change arises and it is necessary to recognize new network configuration by the change in the number of nodes by the insert and remove of a node, ON/OFF of a power source, etc.), each node which detected change transmits a bus reset signal on a bus, and goes into the mode in which new network configuration is recognized.
- [0041] The detection approach of the change at this time is performed by detecting change of the bias voltage on a 1394 port base.
- [0042] A bus reset signal is transmitted from a certain node, BASURI set occurrence is transmitted to a link layer, and the physical layer of each node transmits a bus reset signal to other nodes at the same time it receives this bus reset signal. Bus reset is started after all nodes finally detect a bus reset signal.
- [0043] Bus reset is started also by giving direct instruction to a physical layer by cable **** which was described previously, starting by the hard detection by the abnormalities in a network etc., the host control from a protocol, etc. Moreover, if bus reset starts, data transfer will be interrupted temporarily, and data transfer in the meantime is kept waiting, and is resumed under new network configuration after termination.
- [0044] The above is the sequence of bus reset.
- [0045] The sequence of node ID decision is explained.
- [0046] After bus reset, each node starts the actuation which gives ID to each node, in order to build new network configuration. The general sequence to the node ID decision from bus reset at this time is explained using drawing 6 and the flow chart of 7 and 8.
- [0047] The flow chart of drawing 6 has shown the activity of a series of buses until it determines Node ID from BASURI set occurrence and can perform data transfer.
- [0048] First, if it is monitoring continuously that bus reset occurs and bus reset occurs in power-source ON/OFF of a node etc. in a network as step S101 here, it will move to step S102.
- [0049] At step S102, in order to know the connection situation of a new network from the condition that the network was reset, declaration of a parentage is made between each node by which direct continuation is carried out. As step S103, if a parentage is determined among all nodes, the one root will be determined as step S104. The parentage of step S102 is declared and the root is not determined, either, until it determines a parentage among all nodes.
- [0050] If the root is determined at step S104, a setup of the node ID to which ID is given to each node next as step S105 will be performed. Since new network configuration will have been recognized in all nodes in order

of a predetermined node if a setup is performed repeatedly and it finally finished setting ID as all nodes as step S106 until a setup of Node ID was performed and ID was given to all nodes, it will be in the condition that data transfer between nodes can be performed as step S107, and data transfer will be started.

[0051] If it will be in the condition of this step S107, it goes into the mode which supervises that bus reset occurs again, and if bus reset occurs, a setup from step S101 to step S106 will be performed repeatedly.

[0052] Although the above is explanation of the flow chart of drawing 6, what expressed the part from bus reset of the flow chart of drawing 6 to root decision and the procedure from after root decision to ID setting termination to the flow chart Fig. in more detail is shown in drawing 7 and drawing 8, respectively.

[0053] First, the flow chart of drawing 7 is explained.

[0054] If bus reset occurs as step S201, network configuration will once be reset. In addition, it is always supervising that bus reset occurs as step S201.

[0055] Next, the flag which shows that it is a leaf (node) to each device as a first stage story of the activity which has a new appreciation of the connection situation of the reset network as step S202 is set. Furthermore, it investigates how many the port which he has [each device] as step S203 is connected with other nodes.

[0056] In order to begin declaration of a parentage after this according to the result of the number of ports of step S204, the number of undefined (parentage is not determined) ports is investigated. immediately after bus reset -- number of ports = -- although it is the number of undefined ports, the number of the undefined ports which are alike and follow and are detected at step S204 with which the parentage is determined changes.

[0057] First, it is restricted to the leaf immediately after bus reset that a parentage can be declared first. It can know that it is a leaf by the check of the number of ports of step S203. As step S205, to the node connected to itself, a leaf is announced "He is a child and partners are parents", and ends actuation.

[0058] Since it is undefined port number >1 at step S204 immediately after bus reset, it moves to step S206 and a flag called a branch is set first, and it waits for the node which the number of ports has recognized to be a branch with two or more at step S203 in order to receive "parents" by parentage declaration from a leaf at step S207.

[0059] It becomes possible to declare "he being a child" to the node connected to the port which remains if the branch in which the leaf declared the parentage to and received it at step S207 checks the number of undefined ports of step S204 suitably and the number of undefined ports has become 1. [of step S205] After the 2nd times, it waits in order to receive the "parents" from a leaf or other branches at step S207 again to a certain branch two or more, even if it checks the number of undefined ports at step S204.

[0060] Finally, when a leaf becomes zero on any one branch or an exception target as a result of the number of undefined ports of step S204 (although child declaration could be made, since it did not operate quickly) Declaration of the parentage of the whole network is completed now, the flag of the root is set as step S208, and, as for the only node from which the number of undefined ports became zero (considering all as parents' port decision), the recognition as the root is made as step S209.

[0061] Thus, from the bus reset shown in drawing 7 to declaration of the parentage in a network between all nodes is completed.

[0062] Below, the flow chart of drawing 8 is explained.

[0063] First, since the information on the flag of each node called a leaf, a branch, and the root is set up by the sequence to drawing 8, it carries out based on this and classifies according to step S301, respectively.

[0064] As an activity which gives ID to each node, it is from a leaf that ID can be set up first. A leaf -> branch -> a setup of ID is made from the young number (node number =0-) in order of the root.

[0065] Several Ns (N is the natural number) of the leaf which exists in a network as step S302 are set up.

[0066] Then, it is required that a leaf should give ID to the root each one as step S303. When there are two or more these demands, the root performs an Arbitration as step S304, gives an ID number to one node which won as step S305, and notifies to a negative beam node as a result of failure.

[0067] The leaf which ID acquisition finished with failure as step S306 advances ID request again, and repeats the same activity. As step S307, ID information on the node is transmitted to all nodes by broadcasting from the leaf which has acquired ID. After broadcasting of 1 node ID information finishes, the one number of the remaining leaves is reduced as step S308.

[0068] Here, if the number of these remaining leaves carries out [activity / of the ID request of step S303] as step S309 at a certain time one or more and all leaves finally broadcast ID information, step S309 will be set to

$N=0$, and, next, it will move to ID setup of a branch. ID setup of a branch as well as the time of a leaf is performed.

[0069] First, several M (M is the natural number) of the branch which exists in a network as step S310 is set up.

[0070] Then, it is required that a branch should give ID to the root each one as step S311. On the other hand, as step S312, the root performs an Arbitration and gives it from the following young number which it finished giving to a leaf sequentially from the branch which won.

[0071] The branch which the root notified ID information or a failure result to the branch which advanced the demand as step S313, and ID acquisition finished with failure as step S314 advances ID request again, and repeats the same activity.

[0072] As step S315, ID information on the node is transmitted to all nodes by broadcasting from the branch which has acquired ID.

[0073] After broadcasting of 1 node ID information finishes, the one number of the remaining branches is reduced as step S316.

[0074] Here, as step S317, one or more, at a certain time, finally the number of these remaining branches is performed [activity / of the ID request of step S311], until all branches broadcast ID information. If all branches acquire Node ID, step S317 will be set to $M=0$, and will also end ID acquisition mode of a branch.

[0075] Since the node which finally does not acquire ID information is only the root after ending so far, the youngest number is set up with its ID number by the number which has not been given as step S318, and ID information on the root is broadcast as step S319.

[0076] As shown in drawing 8, after determining a parentage above, a procedure until ID of all nodes is set up is completed.

[0077] Next, the actuation in the actual network shown in drawing 9 as an example is explained, referring to drawing 9.

[0078] As explanation of drawing 9, direct continuation of Node A and the node C is carried out to the low order of Node (root) B, direct continuation of the node D is further carried out to the low order of Node C, and it has a layered structure by which direct continuation of Node E and the node F was further carried out to the low order of Node D. The procedure of determining this layered structure and root node, and Node ID is explained below.

[0079] After bus reset is carried out, in order to recognize the connection situation of each node first, declaration of a parentage is made between the ports where direct continuation of each node is carried out. With this parent and child, a parents side can say that it becomes a high order by the layered structure, and a child side serves as low order.

[0080] At drawing 9, Node A declared the parentage to the beginning after bus reset. A parentage can be declared from the node (it is called a leaf) which has connection only in one port of a node fundamentally. Since this can know only connection of one port first for itself, it recognizes that it is a network edge by this, and the parentage is determined from the node which operated early in it. In this way, the port of the side (between A-B the node A) which declared the parentage is set up with a child, and the port of the other party (node B) is set up with parents. In this way, between node A-B, it is decided between child-parents and node F-D between child-parents and node E-D that they will be child-parents.

[0081] the first [further] floor layer -- it goes up and the parentage is shortly declared to the high order further one by one from what received declaration of the parentage from other nodes among nodes (it is called a branch) with two or more connection ports. In drawing 12, first, Node D is declaring the parentage to Node C, after determining a parentage between D-E and between D-F, and as a result, it is decided between node D-C that they will be child-parents.

[0082] The node C which received declaration of the parentage from Node D is declaring the parentage to the node B connected to another port. It is decided between node C-B by this that they will be child-parents.

[0083] Thus, a layered structure like drawing 9 was constituted and the node B which became parents in all the ports finally connected was determined as the root node. As for the root, only one exists during one network configuration.

[0084] In addition, although Node B was determined as the root node in this drawing 9, if the node B in which this received parentage declaration from Node A is making parentage declaration to early timing to other nodes,

it has moved from the root node to other nodes. That is, depending on the timing transmitted, every node may turn into a root node, and a root node is not necessarily regularity in the same network configuration, either.

[0085] If a root node is determined, next, it will go into the mode in which each node ID is determined. Here, all nodes notify their node ID to all other nodes (broadcasting function). [who determined]

[0086] Self-ID information includes the information on its node number, the information on the location connected, the number of the ports which it has, the number of ports with connection, and the parentage of each port etc.

[0087] As a procedure of assignment of a node ID number, it can start from the node (leaf) which has connection only in one port first, and is assigned with node number =0, 1 and 2, and ... sequentially from this inside.

[0088] The node which gained Node ID transmits the information containing a node number to each node by broadcasting. It is recognized by this that the ID number is "finishing [assignment]."

[0089] If all leaves finish acquiring the self-node ID, the node ID number which next moved to the branch and followed the leaf will be assigned to each node. Node ID information is broadcast one by one like a leaf from the branch to which the node ID number was assigned, and, finally a root node broadcasts self-ID information. That is, the root always owns the greatest node ID number.

[0090] Assignment of the node ID of the whole layered structure finishes as mentioned above, network configuration is reconstructed, and the initialization activity of a bus is completed.

[0091] An Arbitration is explained.

[0092] By 1394 serial bus, the Arbitration (mediation) of a bus royalty is surely performed in advance of data transfer. Since a 1394 serial bus is a logical bus mold network as each device connected according to the individual tells this signal in a network to all devices by relaying the transmitted signal, respectively, the Arbitration is required of the semantics which prevents the collision of a packet. Only one node can transmit to a certain time amount by this.

[0093] Drawing of bus licence is shown in drawing (b) of a bus use demand as drawing for explaining an Arbitration at drawing 10 (a), and it explains using this below.

[0094] If an Arbitration starts, one or two or more nodes will emit a demand of a bus royalty toward a parent node, respectively. Node C and Node F of drawing 10 (a) are a node which has emitted the demand of a bus royalty. The parent node (drawing 10 the node A) which received this emits a demand of a bus royalty toward a parent node further (it acts as intermediary). This demand is sent to the root which finally arbitrates.

[0095] The root node which received the bus use demand determines whether a bus is made to use it for which node. This mediation can perform only a root node and the licence of a bus is given to the node which won by mediation. In drawing 10 (b), licence is given to Node C and use of Node F is refused drawing.

[0096] An Arbitration is told about delivery and having been refused for DP (data prefix) packet to a negative beam node. The bus use demand of the refused node is kept waiting to a next Arbitration.

[0097] The node which won the Arbitration as mentioned above and obtained the licence of a bus can start a data transfer henceforth.

[0098] Here, a series of flow of an Arbitration is shown and explained to flow chart drawing 11 .

[0099] In order for a node to be able to start data transfer, it is required for a bus to be an idle state. In order to complete the data transfer currently performed previously and to recognize that a current bus is idle status, it is judged that each node can start a transfer of it by going through the predetermined idle-time gap length (for example, subaction gap) set up according to the individual by each transfer mode.

[0100] It judges whether as step S401, the predetermined gap length according to data transmitted, respectively, such as Async data and Iso data, was obtained. Since a demand of a bus royalty required in order to start a transfer cannot be performed unless predetermined gap length is obtained, it waits until predetermined gap length is obtained.

[0101] If gap length predetermined at step S401 is obtained, it judges whether there are any data which should be transmitted as step S402, and in order to transmit as step S403, in a certain case, a demand of a bus royalty will be emitted to the root so that a bus may be secured. Finally transfer of the signal showing the demand of a bus royalty at this time is sent to the root, relaying each [in a network] device, as shown in drawing 13 . When there are no data transmitted at step S402, it stands by as it is.

[0102] Next, as step S404, if or more one root receives the bus use demand of step S403, the root will

investigate the number of nodes which advanced the use demand as step S405.

[0103] the selection value in step S405 -- the number of nodes -- when it is =1 (the node which advanced the royalty demand is one), the next bus licence will be given to the node. If the selection value in step S405 is node number >1 (the node which advanced the use demand is plurality), the root will perform mediation which determines the node which gives licence as step S406 as one. It has the composition that the right is equally granted so that this mediation may be fair and only the nodes same each time may not obtain authorization (fair Arbitration).

[0104] Selection divided into one node which the root arbitrated as step S407 out of two or more nodes which advanced the use demand by step S406, and obtained licence, and the node of others which were beaten is performed. From the selection value of step S405 to one node which was arbitrated and obtained licence here, or the number of use demand nodes = the root sends an enabling signal to the node which obtained licence without mediation by 1 to the node as step S408.

[0105] The node which acquired the enabling signal carries out transfer initiation of the data (packet) which should be transmitted immediately after receiving. Moreover, mediation of step S406 is lost, DP (data prefix) packet which shows Arbitration failure is sent to the node to which bus use was not permitted from the root as step S409, and the node which received this stands by until return and predetermined gap length are obtained to step S401, in order to advance the bus use demand for transmitting again.

[0106] The above is explanation of flow chart drawing 11 explaining the flow of an Arbitration.

[0107] An Asynchronous (asynchronous) transfer is explained.

[0108] An asynchronous transfer is asynchronous transmission. The time transition state in an asynchronous transfer is shown in drawing 12.

[0109] The subaction gap of the beginning of drawing 12 shows the idle state of a bus. When this idle time becomes constant value, the node which wishes to transmit judges that a bus can be used and performs the Arbitration for bus acquisition.

[0110] If the licence of a bus is obtained by the Arbitration, a data transfer will be performed in a packet format next. The node which received is ask after data transfer about ask of a receiving result to the transmitted data (return code for the confirmation of receipt). A transfer is completed by returning, answering or sending a response packet after a short gap called gap. ask consists of 4 bits information and a 4-bit checksum, and a transmitting agency node is immediately returned including the information whether they are a success, a busy condition, and pending status.

[0111] Below, the example of the packet format of an asynchronous transfer is shown in drawing 13.

[0112] There are data division and a header unit other than the data CRC for error corrections, the purpose node ID, the source node ID, transfer data die length, various codes, etc. as shown in drawing 13 are written in the header unit, and a transfer is performed in a packet.

[0113] Moreover, an asynchronous transfer is the communication link of 1 to 1 to a partner node from a self-node. Although the packet transmitted from the source node spreads round each node in a network, since things other than the address addressed to themselves are disregarded, only one node of the destination will read.

[0114] The above is explanation of an asynchronous transfer.

[0115] An Isochronous (synchronization) transfer is explained.

[0116] An isochronous transfer is synchronous transmission. Especially this isochronous transfer that can be said to be the greatest description of a 1394 serial bus is the transfer mode suitable for the data transfer which needs the real time transfer of image data, multimedia data called voice data.

[0117] Moreover, this isochronous transfer is uniformly transmitted to all other nodes from one node of the source by the broadcasting function to the asynchronous transfer (asynchronous) having been a transfer of 1 to 1.

[0118] Drawing 14 is drawing in an isochronous transfer showing a time transition state.

[0119] An isochronous transfer is performed for every fixed-on bus time amount. This time interval is called an isochronous cycle. the isochronous cycle time -- 125microS it is . The cycle-start packet is bearing the role which shows the start time of each of this cycle and performs timing of each node. The node called a cycle master transmits a cycle-start packet, and after the transfer termination in the cycle in front of one, after passing through a predetermined idle period (subaction gap), it transmits the cycle-start packet which tells initiation of this cycle.

[0120] The time interval to which this cycle-start packet is transmitted is 125microS. It becomes.

[0121] Moreover, as it was indicated in drawing 14 as Channel A, Channel B, and Channel C, when two or more sorts of packets can give Channel ID into 1 cycle, respectively, it can distinguish and transmit. By this, the real time transfer between two or more nodes is possible to coincidence, and he incorporates only the data of the channel ID needed by the node which receives. This channel ID does not express the address of a transmission place, and has given the logical number to data. Therefore, transmission of a certain packet will be transmitted by broadcasting which spreads round all other nodes from the transmitting agency node of one.

[0122] In advance of packet transmission of an isochronous transfer, an Arbitration is performed like an asynchronous transfer. However, since it is not the communication link of 1 to 1 like an asynchronous transfer, ask (reply code for the confirmation of receipt) does not exist in an isochronous transfer.

[0123] Moreover, iso shown in drawing 14 gap (isochronous gap) expresses the idle period required in order to recognize it as a bus being idle status before performing an isochronous transfer. If this predetermined idle period passes, it can judge that the bus is vacant as for a node to perform an isochronous transfer, and the Arbitration before a transfer can be performed.

[0124] Below, the example of the packet format of an isochronous transfer is shown in drawing 15 , and it explains to it.

[0125] The transfer data length as there are data division and a header unit other than the data CRC for error corrections in various kinds of packets, respectively and shown in the header unit at drawing 15 and Channel NO which were divided into each channel, other various codes, the header CRC for error corrections, etc. are written in, and a transfer is performed.

[0126] The above is explanation of an isochronous transfer.

[0127] A bus cycle is explained.

[0128] An isochronous transfer and an asynchronous transfer can be intermingled in the transfer on an actual 1394 serial bus. Drawing showing the signs of time transition of the transfer condition on a bus that the isochronous transfer and asynchronous transfer at that time were intermingled is shown in drawing 16 .

[0129] Priority is given to an isochronous transfer over an asynchronous transfer, and it is performed. The reason is gap length (isochronous gap) shorter than the gap length (subaction gap) of an idle period required in order to start an asynchronous transfer after a cycle-start packet, and is because an isochronous transfer can be started. Therefore, an isochronous transfer will be given priority to and performed from an asynchronous transfer.

[0130] In the general bus cycle shown in drawing 16 , a cycle-start packet is transmitted to each node from a cycle master at the time of the start of cycle #m. By this, after performing time-of-day adjustment by each node and waiting for a predetermined idle period (isochronous gap), the node which should perform an isochronous transfer performs an Arbitration and starts a packet transfer. In drawing 16 , the isochronous transfer of Channel e, Channel s, and the channel k is carried out at order.

[0131] An asynchronous transfer can be performed, if all isochronous transfers in cycle #m are completed after performing actuation from this Arbitration to a packet transfer repeatedly by the channel to which it is given.

[0132] When the idle time reaches the subaction gap which can be transmitted asynchronous, it is judged that it can move from a node to perform an asynchronous transfer to activation of an Arbitration.

[0133] However, the period which can perform an asynchronous transfer is restricted when the subaction gap for starting an asynchronous transfer from after isochronous transfer termination before the time amount (cycle synch) which should transmit the following cycle-start packet is obtained.

[0134] In cycle #m of drawing 16 , 2 packet (packet 1, packet 2) transfer of the asynchronous transfer (included ack) is carried out after that with the isochronous transfer for three channels. Since after this asynchronous packet 2 results in the time amount (cycle synch) which should start a cycle m+1, the transfer by cycle #m is finished even here.

[0135] However, supposing it results in the time amount (cycle synch) which should transmit the following cycle-start packet during asynchronous or synchronous transmission actuation, it is not interrupted by force, but after waiting for the idle period after the transfer is completed, the cycle-start packet of degree cycle will be transmitted. That is, one cycle is 125microS. When it continues above, degree cycle is 125microS of criteria that much. Suppose that it was shortened. Thus, an isochronous cycle is 125microS. It exceeds on criteria and can be shortened on them.

[0136] However, if an isochronous transfer is the ** cycle need in order to maintain a real-time transfer, it may surely be performed, and an asynchronous transfer may be turned to the cycle after a degree by having shortened the cycle time. It is managed by the cycle master also including such delay information.

[0137] Then, the gestalt of the 1st operation is explained first.

[0138] Drawing 17 is drawing which expresses the description of this invention best, if the interface of 1394 is made to contrast with each class of the OSI model used by LAN in this drawing, the physical layer 1 and the data link layer 2 of an OSI model correspond to the physical link layer which is the lower layer 4 of 1394 interfaces, and the upper layer 3 which exists on these lower layers 4 corresponds to the transport protocol layer 5 and the presentation layer 6 in 1394 interfaces. Moreover, the LOGIN protocol 7 used as the description of this invention operates between the lower layer 4 of 1394 interfaces, and the transport protocol 5.

[0139] With the gestalt of this operation, it can notify that he wants to perform picking using the protocol based on SBP-2 to a partner's device by inserting a LOGIN protocol in the device based on the serial bus protocol (SBP-2) 8. Moreover, in Example 2, by inserting a LOGIN protocol also about the device protocol 9 in which it specialized on 1394 interfaces, each other device can distinguish whether the protocol is supported, and data can be exchanged.

[0140] Drawing 18 is drawing having shown basic actuation of a LOGIN protocol, and in case a printer performs the printing task 10, it determines which is chosen and printed among printer protocol A-B-C currently first prepared by the printer using a LOGIN protocol, and performs printing actuation along with the determined protocol after that. That is, in the device which is supporting the printer protocol of some [a printer side], the protocol of a partner's device is first distinguished using a LOGIN protocol in the case of connection with a target, and a printer chooses from plurality one printer protocol doubled with a partner's protocol, performs an exchange of printing data and a command along with this selected protocol, and performs printing processing.

[0141] Drawing 19 is drawing having shown the topology of each device in 1394 interfaces which mounted the LOGIN protocol in the gestalt of this operation. When the device (PC12, a scanner 13, VCR14 grade) which mounted the LOGIN protocol to the printer 11 corresponding to two or more printer protocols is connected By changing a printer protocol by the printer side according to a partner's transport protocol using a LOGIN protocol, it becomes possible to process the printing task from each device satisfactory.

[0142] Drawing 20 shows the flow of log in actuation.

[0143] First step: Acquisition or the capability to cut is stored [capability / (protocol to receive) / of a lock printer] in the register 503 mentioned later in - device (multi-protocol printer in this case).

- a host's capability -- a printer -- second step [of a set]: -- the protocol determined by - first step -- third step [of a communication link]: and printer, and host of print data -- a connection -- cutting [0144] Drawing 21 can be set in the gestalt of this operation, CRC of the 1394 serial bus with which a device is equipped for a LOGIN protocol is shown, in 501 in drawing, a lock register (lock) and 502 show a protocol register (protocol), and 503 shows a capability register (capability). These registers are arranged to the address set to the initial unit space in the address space of a 1394 serial bus.

[0145] The lock register 501 expresses the lock condition of a resource, and means having already logged in in the state of a lock except "0."

[0146] The capability register 503 expresses the protocol which can be set up and each bit supports each protocol. The value of a bit means that the protocol can be set up in "1", and means that it cannot set up by "0."

[0147] The value of the bit which the protocol register 502 expresses the protocol by which a current setup is carried out, and is equivalent to the bit of the capability register 503 corresponding to the set-up protocol is set to "1."

[0148] Moreover, there are h bits which shows that it is a host device, t bits which shows that it is a target device, p bits which shows that it is a protocol master, and m bits which shows that it is a protocol corresponding to a multi-log in in the protocol register 502 besides the bit which shows a protocol.

[0149] There is a reserved bit corresponding to h bits which shows that it can become a host device other than the bit which shows the protocol which can be set up to the capability register 503, t bits which shows that it can become a target device, p bits which shows that it can become a protocol master, and m bits of the protocol register 502.

[0150] It is the device which acquires the protocol which can set up the device which supports the LOGIN

protocol connected to the 1394 serial bus, takes majority, and notifies to each device that the protocol which should be used giving priority is a protocol master here. When two or more devices which can become a protocol master exist on a 1394 serial bus, a protocol master is determined by adjusting between nodes. [0151] Drawing 22 is a flow chart which shows the majority processing of a protocol which a protocol master performs.

[0152] First, initialization processing for the total of a protocol is performed (step S801).

[0153] Next, it judges whether the information on all nodes was acquired (step S802).

[0154] When no total of nodes is completed as a result of the judgment of step S802, the contents of the capability register 502 of the node to which its attention is paid are read, and the protocol which can be set up is acquired (step S803).

[0155] And for every protocol which can be set up, a node is totaled (step S804) and it returns to step S802 for processing of the following node.

[0156] When the total of all nodes is completed as a result of the judgment of step S802 when the total of all nodes was completed namely, initialization processing for a notice is performed (step S805).

[0157] Next, it judges whether it notified to all nodes (step S806).

[0158] This processing is ended when the notice to all nodes is completed as a result of the judgment of step S806.

[0159] On the other hand, when no notice to nodes is completed, a priority protocol is notified (step S807) and it returns to step S806 for processing of the following node.

[0160] Drawing 23 is the flow chart which showed the log in processing by the side of a host.

[0161] In order to start a log in, the contents of the lock register 501 of the printer (target) which is going to log in, the protocol register 502, and the capability register 503 are first acquired by the lead transaction (step S600).

[0162] Next, it distinguishes whether it is that the printer is supporting the protocol which a host uses for a communication link (log in processing initiation O.K.) from the contents of the capability register 503 obtained at step S600 (step S601).

[0163] This processing is ended when the printer is not supporting the protocol which a host uses for a communication link by the distinction result of step S601. Thereby, this log in processing is stopped.

[0164] On the other hand, when the printer is supporting the protocol which a host uses for a communication link, it distinguishes whether it has already logged in to the resource of a printer in the state of the lock from the contents of the lock register 501 obtained at step S600 because the value distinguishes whether it is "0" (step S602).

[0165] By the distinction result of step S602, when a printer is logging in with other devices (lock register 501! =0), the protocol by which a current setup is carried out distinguishes whether it is multi-log in correspondence from the contents of the protocol register 502 obtained at step S600 (step S608).

[0166] It progresses to step S606 later mentioned when it is multi-log in correspondence by the result of distinction of step S608, and this processing is ended when it is not multi-log in correspondence.

[0167] a printer is not logging in with other devices, and by the distinction result of step S602, when it is possible (lock register 501= 0) in a log in, resource lock processing can be performed, and by things, a lock transaction is used for the lock register 501 of a printer, "1" is written in, and it considers as a log in setup by the side of a host (step S603). It means that the printer had been locked by this step S603, and modification of the contents of a register from other devices will be in an impossible condition by it.

[0168] If it will be in the condition that the resource of a printer was locked next, a protocol will be set up, but in order to support two or more printer protocols, the printer in the gestalt of this operation needs to determine the protocol used between host sides, before receiving print data.

[0169] So, a host side gives priority to this here, if a host and the both sides of a printer support the protocol notified from the protocol master mentioned above. Moreover, priority is given to this if the protocol corresponding to a multi-log in is in the protocol which a host and the both sides of a printer support.

[0170] Thus, if the protocol to be used is determined, this protocol will distinguish whether it is the protocol set as the printer (step S604).

[0171] When the determined protocol and the protocol set as the printer change with results of distinction of step S604, a protocol is changed by performing protocol setting processing (step S605). This protocol setting

processing is processing in which a light transaction sets up and notifies the bit equivalent to the protocol to be used from now on to "1" to the protocol register 502 of a printer.

[0172] The host who the protocol which a host uses for a communication link is notified to a printer at this time, and is doing the current log in since a printer is in a lock condition transmits print data with the usual protocol (step S606). Moreover, when a log in is required, it logs in to the protocol used for a communication link with the protocol. Furthermore, also when the protocol is multi-log in correspondence, it logs in here.

[0173] And after transmission of print data is completed, log out processing is performed (step S607), and this processing is ended. That is, a host is clearing the lock register 501 of a printer, and cancels the resource lock of a printer. Moreover, when communicating with the protocol corresponding to a multi-log in and other devices do not log in, the resource lock of a printer is canceled.

[0174] Drawing 24 is the flow chart which showed the log in processing by the side of a print (target side).

[0175] A printer is usually in a log in waiting state from a host (step S701). Since the print request from a host is started by reading of the lock register 501 of a printer, the protocol register 502, and the capability register 503 at this time, these registers are always changed into the condition which can be read from other devices.

[0176] Then, the host who is going to perform print-out presupposes now that the printer was locked in writing in to the lock register 501 of a printer.

[0177] A lock of a printer distinguishes the notice of resetting of the protocol used from a host first (step S702).

[0178] By the result of distinction of step S702, when there is a notice of a protocol, a printer switches the protocol which he receives, resets a protocol according to a host side (step S703), and progresses to the following step S704.

[0179] By the result of distinction of step S702, when there is no notice of a protocol, it progresses to step S704, without performing processing of step S703.

[0180] Next, step S704 - step S707 perform the communication link by the protocol used. Here, in the protocol corresponding to a multi-log in, the log in by two or more devices in this protocol is received.

[0181] And after all communication links are completed, a printer checks that the lock register 501 has been cleared by the host, and returns to a log in waiting state (step S701) again (step S708).

[0182] In addition, even if it applies this invention to the system which consists of two or more devices as shown in drawing 19, it may be applied to the data-processing approach in the equipment which consists of one device.

[0183] Moreover, it cannot be overemphasized by the purpose of this invention supplying the storage which memorized the program code of the software which realizes the host of the gestalt of each operation mentioned above, and the function of a terminal to a system or equipment, and reading and performing the program code with which the computer (or CPU and MPU) of the system or equipment was stored in the storage that it is attained.

[0184] In this case, the function of the gestalt of each operation which the program code itself read from the storage mentioned above will be realized, and the storage which memorized that program code will constitute this invention.

[0185] As a storage for supplying a program code, a floppy disk, a hard disk, an optical disk, a magneto-optic disk, CD-ROM, CD-R, a magnetic tape, the memory card of a non-volatile, ROM, etc. can be used, for example.

[0186] Moreover, it cannot be overemphasized by performing the program code which the computer read that it is contained also when the function of the gestalt of operation mentioned above is not only realized, but it performs a part or all of processing that OS which is working on a computer is actual, based on directions of the program code and the function of the gestalt of operation is realized by the processing.

[0187] Furthermore, after the program code read from a storage is written in the memory with which the functional expansion unit connected to the extension board inserted in the computer or a computer is equipped, it cannot be overemphasized that it is contained also when the function of the gestalt of operation which performed a part or all of processing that CPU with which the functional add-in board and functional expansion unit are equipped based on directions of the program code is actual, and mentioned above by the processing is realized.

[0188]

[Effect of the Invention] As explained above, according to this invention, the high data-processing approach, the

data processor, printer, and storage of expandability can be offered. When using it from two or more protocols especially, choosing one protocol, the frequency of modification of a protocol can be made small and decline in the effectiveness by resetting of a protocol can be prevented. Moreover, occupancy of a device can be prevented by giving priority to the protocol which permits the log in of two or more devices.

[Translation done.]

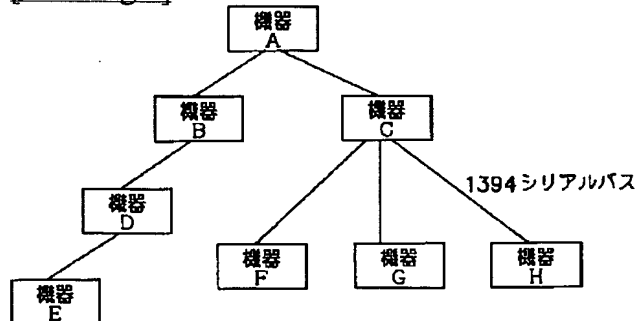
* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

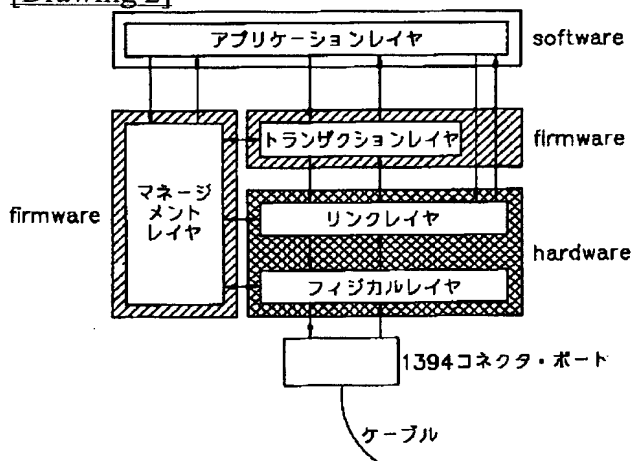
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DRAWINGS

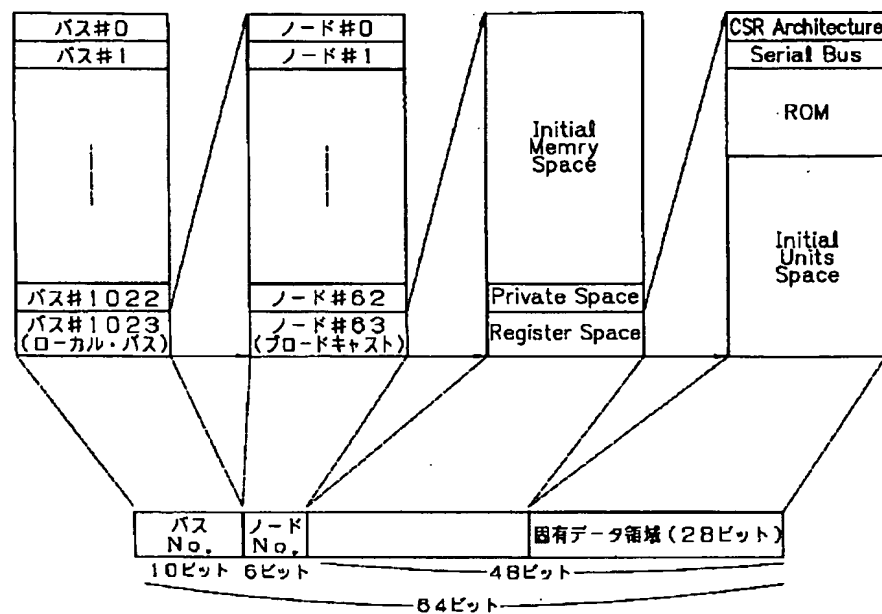
[Drawing 1]



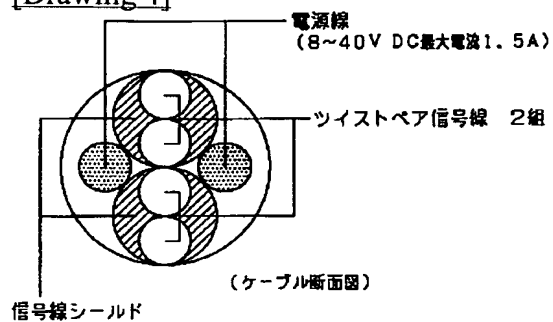
[Drawing 2]



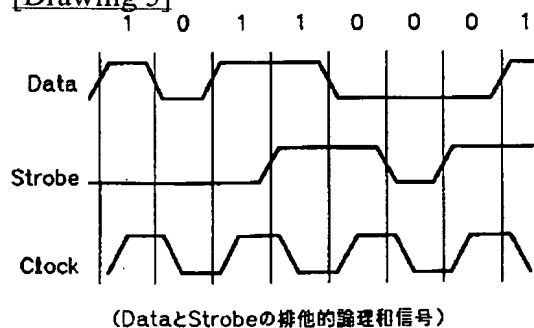
[Drawing 3]



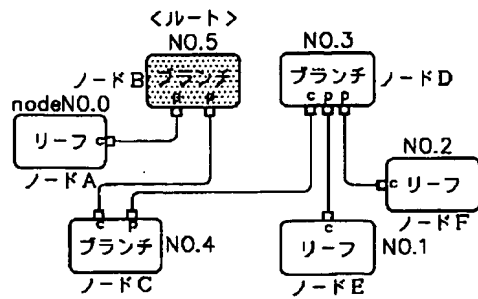
[Drawing 4]



[Drawing 5]



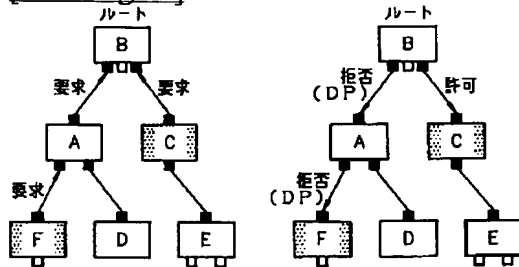
[Drawing 9]



ブランチ：2つ以上のノード接続があるノード
 リーフ：1つのポートのみ接続があるノード

□：ポート
 c：子のノードに相当するポート
 p：親のノードに相当するポート

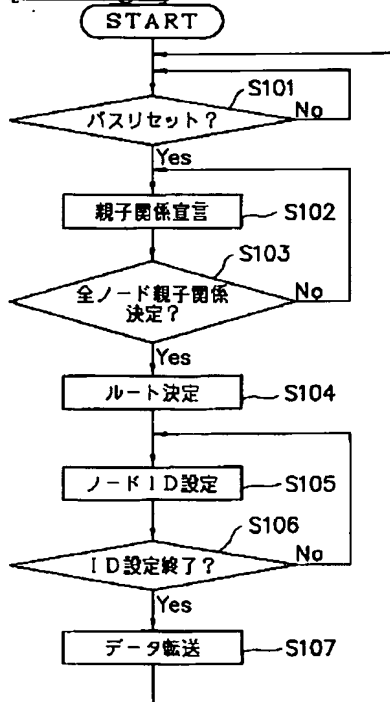
[Drawing 10]



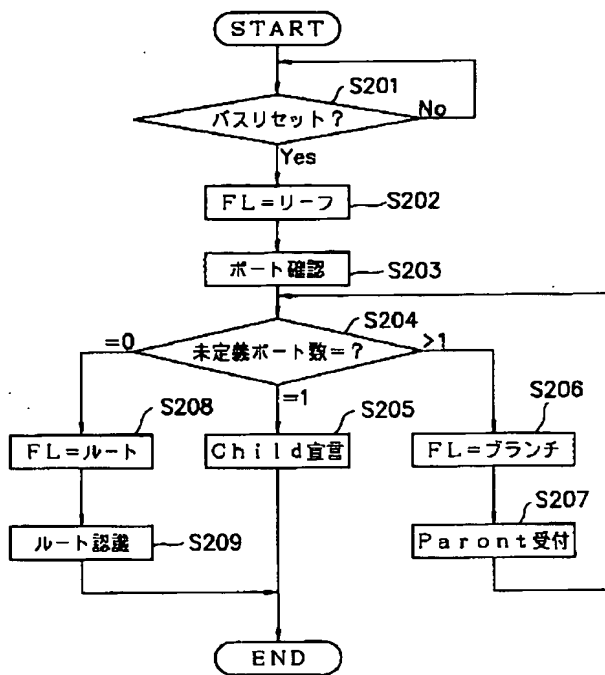
(a) バス使用権の要求

(b) バス使用の許可

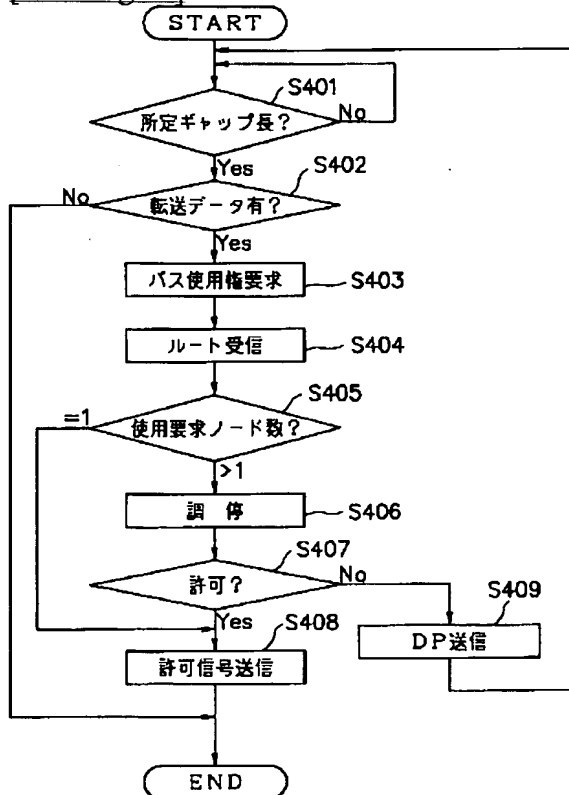
[Drawing 6]



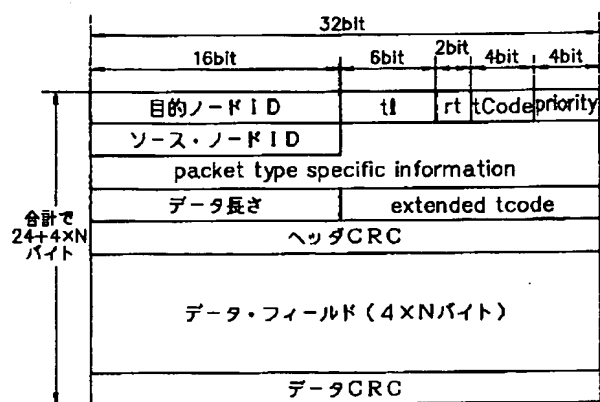
[Drawing 7]



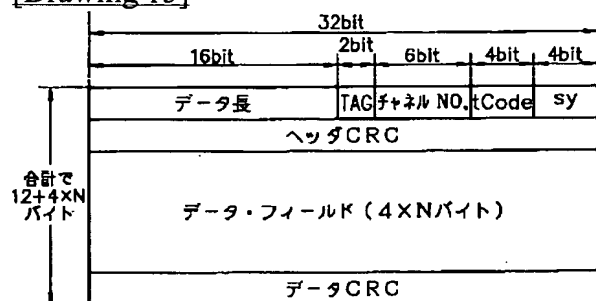
[Drawing 11]



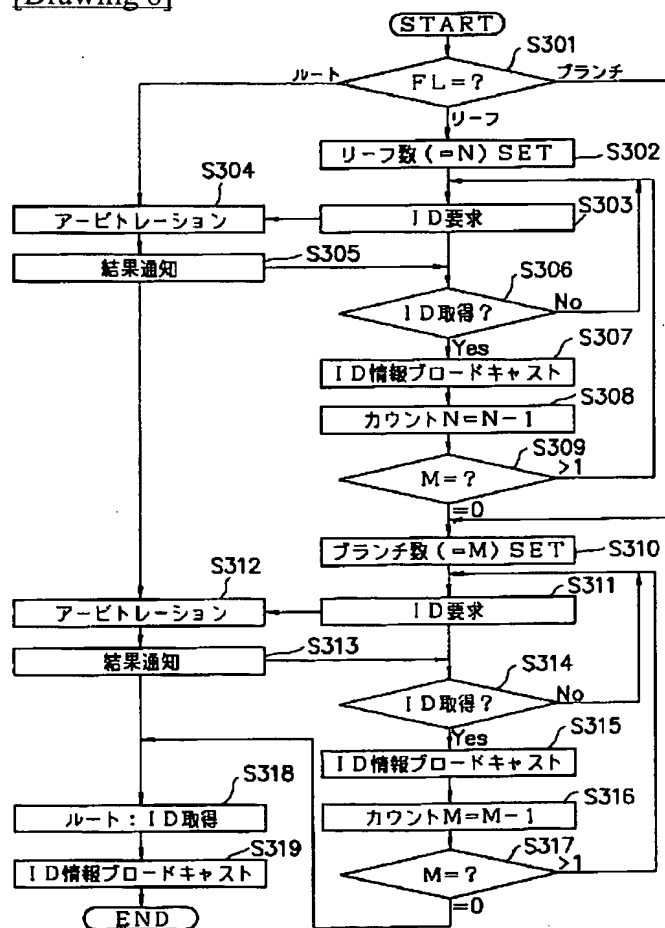
[Drawing 13]



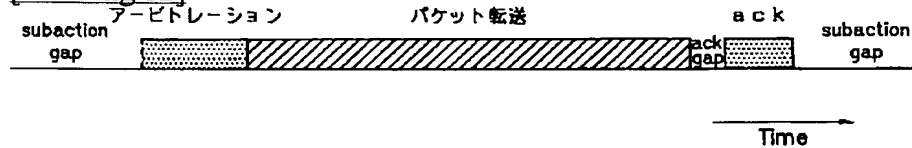
[Drawing 15]



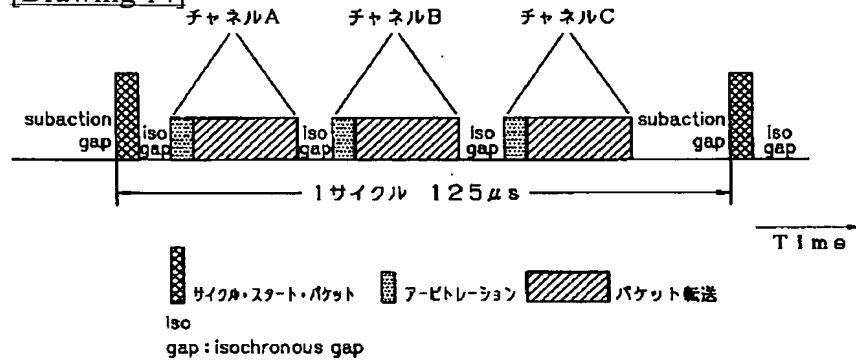
[Drawing 8]



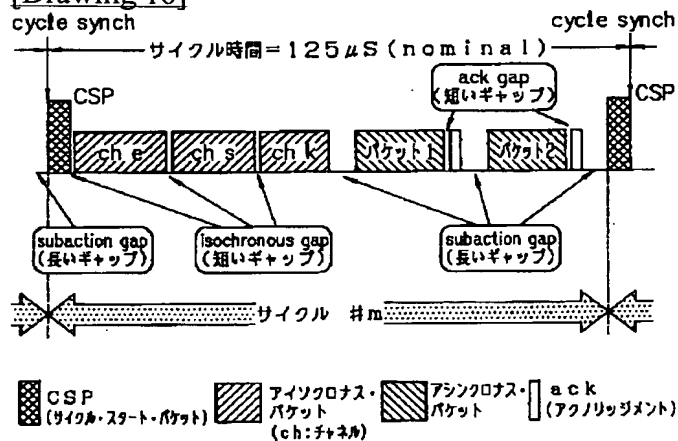
[Drawing 12]



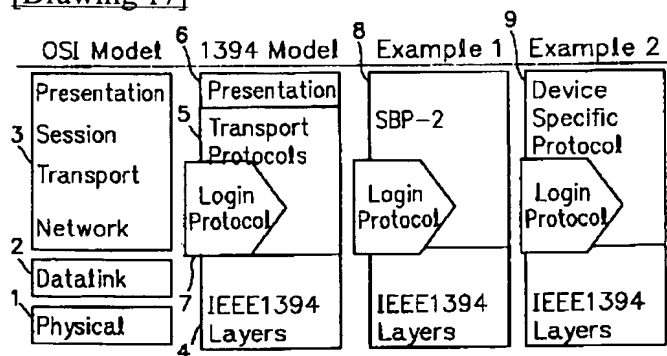
[Drawing 14]



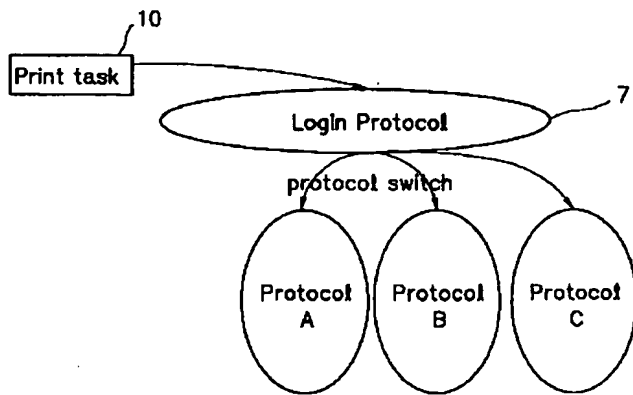
[Drawing 16]



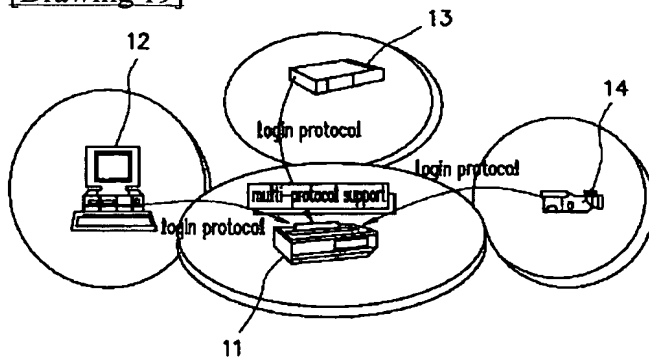
[Drawing 17]



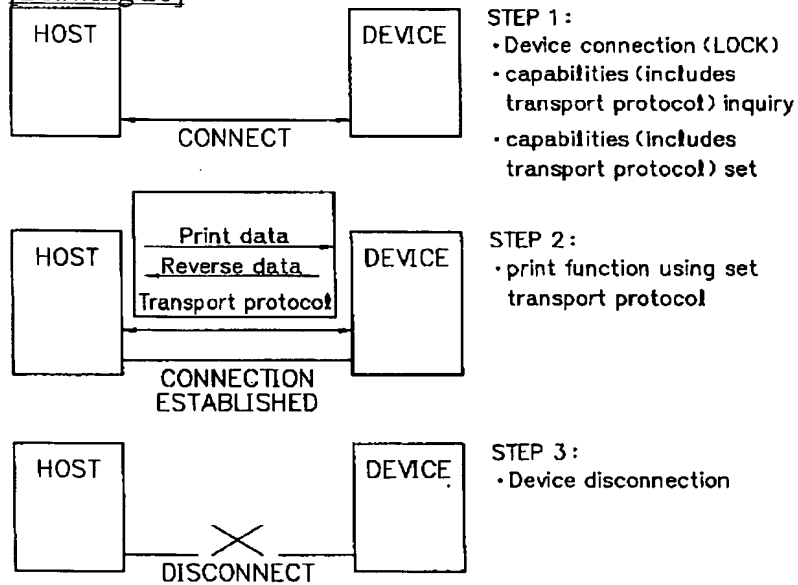
[Drawing 18]



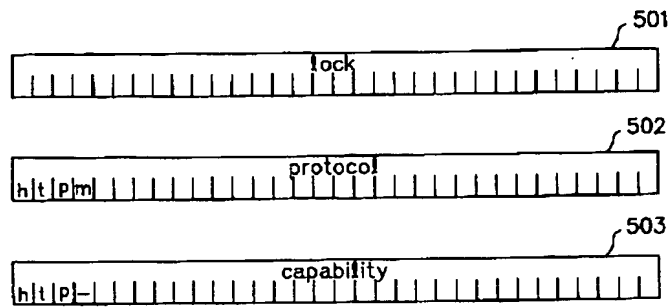
[Drawing 19]



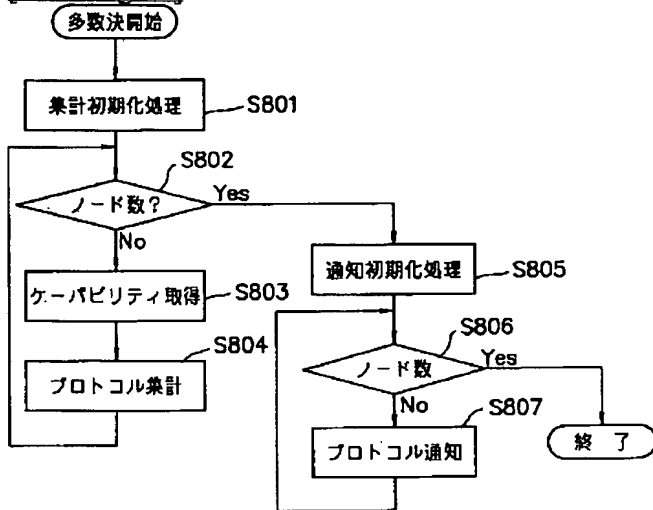
[Drawing 20]



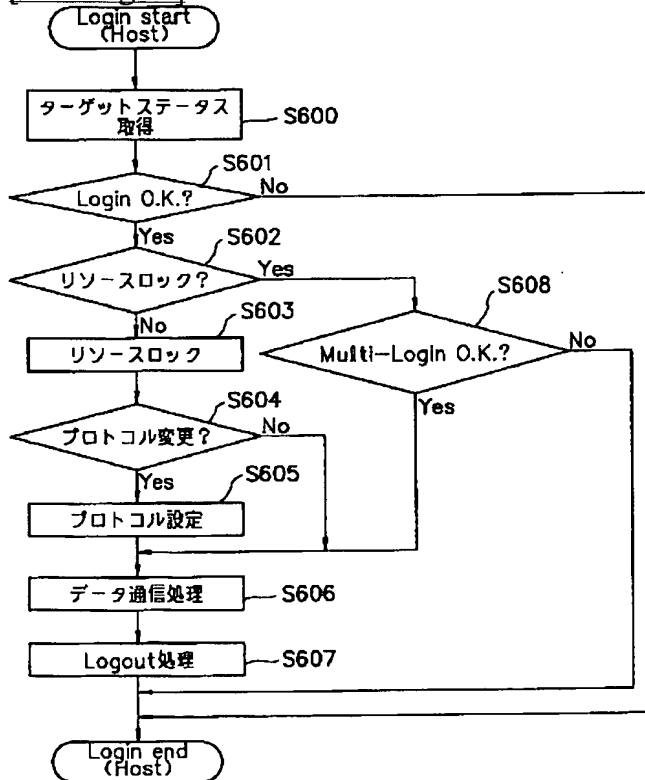
[Drawing 21]



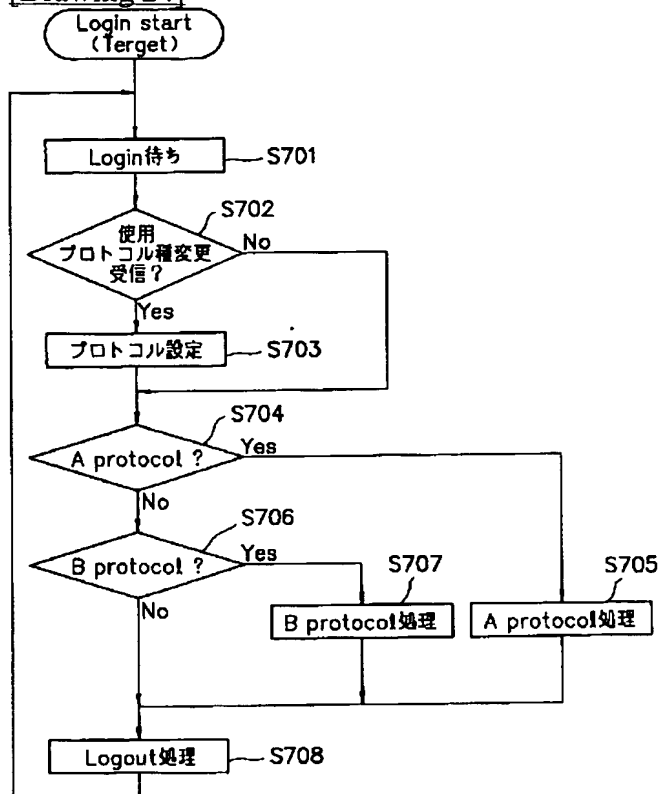
[Drawing 22]



[Drawing 23]



[Drawing 24]



[Translation done.]